# Petrinetzbasierte Validierung von Eisenbahnsicherungssystemen

## Validation d'automatismes ferroviaires de sécurité à base de réseaux de Petri

Von der Fakultät für Maschinenbau der Technischen Universität Carolo-Wilhelmina zu Braunschweig

genehmigte Dissertation

zur Erlangung der Würde

eines Doktor-Ingenieurs (Dr.-Ing)


| | |
|---|---|
| Von: | Dipl.-Ing. Marc Pierre Joseph Antoni |
| Aus (Geburtsort): | Haguenau (Bas-Rhin) |
| Eingereicht am: | 23.10.2009 |
| Mündliche Prüfung am: | 16.12.2009 |
| Haupt Referenten: | Prof. Dr.-Ing. Dr. h.c. E. Schnieder |
| | Prof. Dr.-Ing. habil. G. Tarnai |
| Prüfungsvorsitz: | Prof. Dr.-Ing. K. Lemmer |


**2009**

# Remerciements

Je tiens tout d'abord à remercier le professeur **Eckehard SCHNIEDER** de l'Institut pour la sécurité des transports et des automatismes de l'université technologique de Braunschweig, mon directeur de thèse qui a bien voulu me donner ma chance et m'assister pour ce travail.

Je tiens à remercier à M. **Jacques COUVERT**, Directeur Délégué Infrastructure de la SNCF pour le soutien qu'il m'a toujours apporté à la réalisation de ce projet au sein de notre Entreprise, soutien sans lequel la présente thèse n'aurait jamais été possible.

Mes remerciements vont également à M. **Frank BERNARD**, Directeur de la plaque ingénierie Nord Paris, mes amis, pour le soutien permanent, tant moral que managérial, depuis l'origine de notre projet.

Je remercie les personnes du Département « Ingénierie de Maintenance » de l'Infrastructure qui m'ont permis de mener à bien ce travail MM. **Philippe PETIT, Vincent MAUMY, Nadia AMMAD et Patrick MAILLOT**.

Mes remerciements à **Carolina MEIER HIRMER** pour sa patience lors de la correction de mon allemand.

# Danksagung

Ich möchte vor allem meinem Doktorvater Herrn **Prof. Dr. Schnieder** vom Institut für Verkehrssicherheit und Automatisierungstechnik der TU Braunschweig danken. Er hat mir die Möglichkeit gegeben, über dieses Thema zu arbeiten und er hat mich sehr bei dieser Arbeit unterstützt.

Ich möchte Herrn **Jacques Couvert,** ehemaliger Direktor der SNCF Infrastruktur, danken, der es mir ermöglicht hat, dieses Projekt innerhalb der SNCF zu leiten und für die Unterstützung, ohne die diese Doktorarbeit nie möglich gewesen wäre.

Mein Dank geht auch an Herrn **Frank Bernard,** Verantwortlicher für die Ingenieursabteilung Paris Nord, an meine Freunde, für die beständige Unterstützung, sowohl moralisch als auch auf der Managementebene, von Anfang des Projekts an.

Ich danke den Personen der Abteilung „Wartungsplanung" der Infrastruktur, die mir erlaubt haben, diese Arbeit erfolgreich durchzuführen: **Philippe Petit**, **Vincent Maumy, Nadia Ammad und Patrick Maillot.**

Mein Dank geht auch an **Dr. Carolina Meier-Hirmer** die bei der Verbesserung meines Deutsch sehr geduldig war.

*A mes parents pour les valeurs qu'ils m'ont données…*

An meine Eltern, für die Werte, die sie mir mit auf den Weg gegeben haben…


**Figure 1 - Familie Stürzel - Schaeffer - Schwartz**

*Hochfelden en 1905*

*Mon arrière grand père aux chemins de fer d'Alsace-Lorraine*

Hochfelden 1905

Mein Urgroßvater bei der Eisenbahn von Elsaß-Lothringen

*Quel homme ne voudrait construire un monde meilleur ? Et même si celui-ci ne dure qu'un instant, au moins aura-t-il vécu.*

*Welcher Mensch möchte nicht eine bessere Welt bauen? Und selbst wenn dieser nur einen Augenblick dauert, wird er mindestens gelebt haben.*

*« Nous sommes des nains juchés sur des épaules de géant. Nous voyons ainsi davantage et plus loin qu'eux, non parce que notre vue est plus aigue ou notre taille plus haute, mais parce qu'ils nous portent en l'air et nous élèvent de toute leur hauteur gigantesque ».*

*Bernard de Chartres, XII$^{ème}$ siècle.*

„Wir sind Zwerge, die auf den Schultern von Riesen stehen. Wir sehen so mehr und weiter als sie, aber nicht weil wir besser sehen oder wir größer sind, sondern weil die Riesen uns in der Luft tragen und uns auf ihre riesige Größe hochziehen."

Bernard von Chartres, XII. Jahrhundert.

*«Dans notre métier [la signalisation], le diable est dans les détails »*

*M. Dupuy (ancien PDG de la SNCF)*

„In unserem Beruf [beim Signalsystem] steckt der Teufel in den Details"

Herr Dupuy (früherer Generaldirektor der SNCF)

*« Pour le bénéfice à court terme je ne compromets pas l'avenir »*

*Werner von Siemens*

„Für augenblicklichen Gewinn verkaufe ich die Zukunft nicht"

Werner von Siemens

"Everything that can go wrong sooner or later will go wrong"

"If there's more than one way to do a job and one of those ways will end in disaster, then somebody will do it that way."

Murphy's Law

*Des enseignements toujours d'actualité…*

Immer aktuelle Lehre…

# Sommaire

# Inhaltsverzeichnis

**ANNEXES**



**ANHANG**

# CHAPITRE 1

# Introduction et contexte

Le système ferroviaire repose sur de très nombreux composants techniques qui permettent d'assurer en sécurité et en qualité le transport de voyageurs et de marchandises. Les fonctions assurées par ces composants sont extrêmement variées, qu'il s'agisse d'éléments de l'infrastructure (voie, signalisation, caténaires…) ou du matériel roulant (locomotives, voitures, wagons, rames…). Les systèmes informatisés ou informatiques prennent une part croissante et bientôt majeure dans les systèmes reposant sur des automatismes.

Le fonctionnement sûr et à moindre défaillance de ces automatismes est une condition essentielle pour réaliser un service de qualité et réduire les coûts induits par les dysfonctionnements et les incidents. Ces coûts intègrent :

- les coûts directs résultant du préjudice causé par la défaillance (pertes humaines, perte de revenu, coûts d'intervention pour le dépannage, coûts de remplacement et de remise en exploitation…) ;
- les coûts indirects occasionnés par les mesures préventives de conception et d'entretien pour réduire l'occurrence et les conséquences des défaillances (interruption de trafic nécessitée par les opérations d'entretien ou de modification, application des procédures réglementaires…).

De façon très classique, comme cela se pratique dans l'industrie, pour se prémunir contre le risque de défaillance non sûre[1] des automatismes des installations de sécurité, les différents services de la Société nationale des chemins de fer français (SNCF) en charge de ces systèmes ont adopté et mis en œuvre différentes politiques de maintenance préventive et (re)mise en exploitation.

[1] Panne sûre : une panne sûre n'a pas d'influence sur la sécurité des personnes et des circulations.

# KAPITEL 1

# Einleitung und Übersicht

Das Bahnsystem besteht aus zahlreichen technischen Komponenten, mit Hilfe derer der Personen- und Güterverkehr unter höchster Sicherheit und bester Qualität abgewickelt werden kann. Die Funktionen dieser Komponenten sind äußerst unterschiedlich, wie beispielsweise die der Infrastruktur (Gleis, Signaltechnik, Oberleitungen…) oder die der Fahrzeuge (Lokomotiven, Personenwagen, Güterwagen, Zugverbände…). IT-Systeme sind immer mehr, und nicht unwesentlich an automatisierten Systemen beteiligt.

Die sichere, nahezu fehlerfreie Funktionsweise dieser Systeme ist eine wesentliche Bedingung für einen Bahnbetrieb hoher Qualität zur Reduzierung von Kosten durch Fehlfunktionen und Zwischenfälle. Diese Kosten umfassen:

- direkte Kosten aufgrund des Schadens durch Versagen (Personenschaden, Einnahmeausfälle, Instandsetzungskosten, Ersatzkosten, Wiederinbetriebnahmekosten, usw.)
- indirekte Kosten aufgrund von Entwicklungs- und Wartungsmaßnahmen zur Reduzierung der Häufigkeit und der Schwere der Folgen des Versagens (Verkehrsunterbrechung wegen einer Instandhaltungs- oder Änderungsmaßnahme, Umsetzungsaufwand von Vorschriften, usw.).

Auf ganz herkömmliche Weise – so wie dies auch in der Industrie der Fall ist – haben die verschiedenen Dienststellen der französischen Bahn (*Société Nationale des Chemins de fer Français*/SNCF), die für diese Systeme verantwortlich sind, zur Verringerung des Risikos eines nicht sicheren Versagens[2] automatisierter Sicherungssysteme verschiedene Maßnahmen zur vorbeugenden Instandhaltung und zur (Wieder-) Inbetriebnahme beschlossen und eingeführt.

[2] Ein sicheres Versagen hat keinen Einfluss auf die Sicherheit der Personen und der Züge.

Elles visent à assurer de la façon la plus efficace la disponibilité maximale en exploitation de ces automatismes pour limiter le nombre de pannes sûres et éviter toute panne non sûre pour les installations de sécurité[3].

L'avènement des systèmes informatisés ou informatiques remet en cause les procédures et pratiques antérieures de validation sans que l'application de normes n'offre de réponses vraiment satisfaisantes en relation avec le niveau de sécurité requis. Les installations de sécurité étant en exploitation en permanence, sans interruption (24 heures sur 24, 365 jours par an), aucun taux d'erreur fonctionnelle non sûre n'est acceptable. Cela reviendrait à accepter, parce que ce sont des installations informatiques, une ou plusieurs défaillances non sûres.

La recherche du quasi-zéro défaut dans le développement et la modification des automatismes informatiques des installations de sécurité demande avec les méthodes actuelles, un effort de validation manuelle ou automatique très important, sans véritable garantie de résultat. Le poids de cet effort conduit les constructeurs de matériels ferroviaires et les autorités de sûreté européenne à accepter cette évolution et réduire le niveau sécurité exigé en Europe pour les nouvelles installations informatiques de sécurité interopérables.

Un axe majeur pour établir de nouvelles stratégies plus efficaces est de créer les conditions d'application et de mise en œuvre industrielle de méthodes formelles sur les installations informatiques de sécurité [Bied, 2003]. La démarche est de définir les conditions de conception et de validation formelle des automatismes des installations de sécurité informatiques de manière à conserver la notion d'installations «fail safe» et les niveaux de sécurité antérieurs, d'évaluer objectivement le niveau de sûreté d'un système associant composant logiciel et matériel. Le problème posé étant un problème industriel avec des enjeux de sécurité et des enjeux économiques, il est donc abordé en adoptant une attitude à la fois pragmatique et théorique. La nécessité du pragmatisme s'impose sous plusieurs aspects :

- il nous faut partir du contexte industriel ferroviaire et des problèmes pratiques réels ;
- les propositions que nous sommes amenés à faire doivent impérativement réduire les coûts de réalisation et de maintien en condition opérationnelle sous peine de ne jamais être appliquées en dépit des gains de sécurité qu'elles pourraient apporter ;

Diese dienen der Gewährleistung der maximalen betrieblichen Verfügbarkeit der Sicherungssysteme, der Beschränkung der Anzahl sicherer Pannen und der Verhinderung jeder nicht sicheren Panne einer Sicherheitseinrichtung[4].

Durch das Aufkommen der IT-Systeme wurden die früheren Verfahren und Vorgehensweisen in Frage gestellt, ohne dass die Anwendung von Normen eine zufrieden stellende Antwort auf das geforderte Sicherheitsniveau bietet. Da die Sicherheitseinrichtungen ununterbrochen in Betrieb sind (rund um die Uhr, an 365 Tagen im Jahr), ist keine nicht sichere funktionelle Fehlerrate hinnehmbar: das hieße, dass nicht sichere Versagen akzeptiert werden, nur weil es sich um IT-Einrichtungen handelt.

Das Erzielen von quasi null Fehlern bei der Entwicklung/Änderung zu Sicherheitseinrichtungen gehörenden IT-Systemen erfordert mit den heutigen Verfahren einen sehr großen Aufwand manueller Validierung, ohne jegliche Ergebnissicherheit. Dieser hohe Aufwand verlangt von den Eisenbahnzugführern und den europäischen Sicherheitsbehörden, eine solche Entwicklung zu akzeptieren und das in Europa für neue, interoperable IT-Sicherheitseinrichtungen geforderte Sicherheitsniveau zu reduzieren.

Eine der Möglichkeiten zur Festlegung neuer, effizienter Strategien ist die Schaffung von Voraussetzungen zur industriellen Anwendung / Implementierung formaler Validierungsverfahren auf IT-Sicherheitseinrichtungen [Bied, 2003]. Die Vorgehensweise besteht in der Bestimmung der Bedingungen für die Konzeption, Realisierung und formale Validierung der Automatismen der IT-Sicherheitseinrichtungen, um das Konzept der fail-safe-Einrichtungen und die früheren Sicherheitsniveaus aufrecht zu erhalten. Es handelt sich hierbei um ein industrielles Problem, bei dem sowohl die Sicherheit als auch die Wirtschaftlichkeit zu beachten sind: es muss also sowohl pragmatisch als auch theoretisch fundiert vorgegangen werden. Pragmatismus ist wegen folgender Aspekte notwendig:

- Es ist vom industriellen Bahnkontext und von den tatsächlichen, praktischen Problemen auszugehen.
- Verbesserungsvorschläge müssen unbedingt die Kosten (der Realisierung/Erhaltung des betriebsfähigen Zustandes) berücksichtigen, sonst werden sie trotz der erreichbaren Sicherheitsgewinne nie angewandt.

[3] Dispositifs de sécurité : ils couvrent les installations de signalisation et veillent aux automatismes

[4] Sicherheitseinrichtungen umfassen die Signalanlagen und deren Steuerungssysteme.

- le facteur humain doit être intégré, notamment pour la définition des propriétés de sécurité.

## 1.1 Démarche

La démarche suivante a été retenue :

1. Problématique de la conception et la validation des installations informatiques de sécurité ;
2. Axes de réponse qu'il est envisageable d'apporter à cette problématique ;
3. État de l'art des méthodes de développement actuelles et les mesures de sécurisation du système ferroviaire puis identification des postulats de fonctionnement et des propriétés de sécurité à vérifier par les automatismes des installations de sécurité ;
4. État de l'art des méthodes formelles, analyse de leurs contributions possibles à notre problématique et choix d'une méthode formelle adaptée à notre environnement ferroviaire, définition des conditions pratiques d'application ;
5. Applications de notre méthode à des exemples théoriques simples ;
6. Applications de notre méthode à des cas ferroviaires concrets.

La démarche de ce travail a été de :

- Choisir une méthode de preuve applicable aux réseaux de Petri, adaptée à un cadre industriel pour des applications séquentielles et combinatoires, et couvrant les risques déterministes liés aux erreurs de spécification et de paramétrage. Il est à noter que la méthode, dans un premier temps réalisée manuellement, permet de mettre en exergue d'une part les comportements non sûrs et, d'autre part, les comportements trop restrictifs (surabondants) du système à valider ;
- Identifier des conditions d'automatisation de cette méthode de preuve : prise en compte dès la conception de la machine cible des contraintes indispensables à la mise en œuvre de la méthode retenue, maîtrise de la combinatoire… Il est à noter que la machine cible devra être définie pour répondre aux exigences de validité de la preuve formelle, en l'occurrence la réalisation d'un véritable automate à nombre fini d'états ;

- Menschliche Fehlleistungen müssen berücksichtigt werden, insbesondere im Hinblick auf die zu gewährleistenden Sicherheitseigenschaften.

## 1.1 Vorgehensweise

Der folgende Denkansatz wird angenommen:

1. Problematik der Darstellung der Konzeption und der Validierung von IT-Sicherheitseinrichtungen.
2. Mögliche Lösungen für diese Problematik.
3. Stand der Technik bei der Bahnsystemsicherung; Identifizierung der Anforderungen zur Funktionsweise und der Sicherheitseigenschaften, die von den automatisierten Sicherheitseinrichtungen zu überprüfen sind.
4. Stand der formalen Verfahren, Analyse ihres möglichen Beitrags zur Problematik und Wahl eines an das Bahnumfeld angepassten formalen Verfahrens, Definition der praktischen Anwendungsbedingungen.
5. Anwendung des ausgewählten Verfahrens auf einfache, theoretische Beispiele.
6. Anwendung des ausgewählten Verfahrens auf konkrete Fälle der französischen Eisenbahn.

Diese Arbeit basiert auf folgenden Schritten:

- Wahl einer auf Petrinetze anwendbaren Beweisführung, die auch für die im industriellen Bereich notwendigen aufeinanderfolgenden und kombinierten Anwendungen geeignet ist und die deterministische Risiken bei Spezifikations- und Parametrierungsfehlern abdeckt. Das zunächst manuelle Verfahren erlaubt es, nicht sichere Verhalten einerseits und zu restriktive Systemverhalten andererseits aufzuzeigen.
- Identifizierung der Bedingungen zur Automatisierung dieser Beweisführung. Bereits bei der Konzeption der Zielmaschine sind die Randbedingungen, die bei der Implementierung des gewählten Verfahrens unabdingbar sind und die Beherrschung der kombinatorischen Explosion zu berücksichtigen… Es ist anzumerken, dass die Zielmaschine spezifisch definiert und konzipiert sein muss, um alle notwendigen Ausführungseigenschaften einhalten zu können, die für eine mathematisch korrekte Beweisführung notwendig sind.

- Définir un formalisme d'écriture des propriétés de sécurité accessibles aux utilisateurs 'métier' et définition d'outils complémentaires pour rendre transparents les aspects mathématiques pour les utilisateurs « métiers » ;
- Appliquer sur un poste réel déjà en exploitation et analyse des résultats.

Le domaine du ferroviaire français servira de support ou de domaine d'application à ce travail :

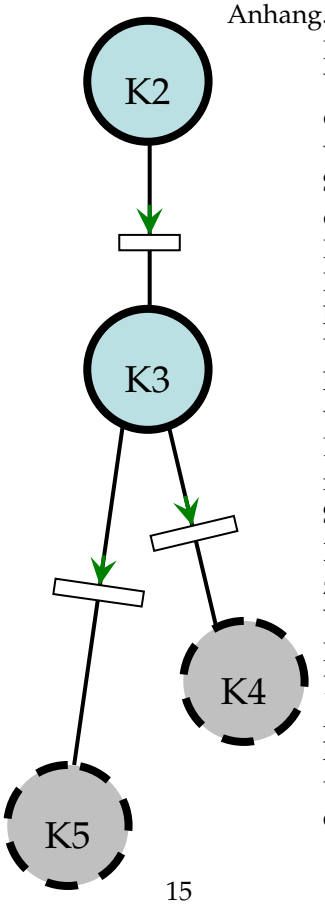
- il bénéficie d'un retour d'expérience de 150 ans, qui a vu plusieurs évolutions technologique majeures (mécanique, électromécanique, électronique, informatique) ;
- c'est un milieu industriel au niveau d'exigence sécurité et disponibilité élevées et une continuité de service à assurer même en mode dégradé (procédures et hommes) ;
- une maîtrise fonctionnelle et technique des installations par les opérateurs (exploitation et maintenance).

## 1.2 Plan du document

Le travail reprend les étapes décrites précédemment et reporte en annexe la description du contexte ferroviaire français, l'étude des différentes méthodes formelles, les détails du traitement des cas d'application.

**Chapitre 2** : Nous présenterons tout d'abord le contexte actuel de la réalisation et de la modification des systèmes informatiques critiques, des difficultés croissantes inhérentes à leur validation avant mise en exploitation. Cette problématique dépasse largement le strict cadre des systèmes critiques ferroviaires. Elle a déjà conduit certains industriels importants et connus à ne pas franchir le pas des systèmes informatiques critiques, voir même à revenir sur les choix antérieurs pour des raisons de maîtrise effective du niveau de sécurité de tels systèmes.

**Chapitre 3** : Nous présenterons les orientations principales proposées pour répondre à cette problématique, des règles de conception des matériels supports aux architectures générales des logiciels de base et des logiciels d'application.




- Definition einer formalen Beschreibungsmethode für Sicherheitseigenschaften, die den Anwendern vor Ort zugänglich ist und Definition von ergänzenden Werkzeugen, um die mathematischen Aspekte für die Anwender transparent zu machen.
- Anwendung auf ein Stellwerk, das bereits betrieben wird, sowie Analyse der Ergebnisse.

Die französische Eisenbahn dient hier als Basis, bzw. als Anwendungsbereich:

- Sie verfügt über 150 Jahre Erfahrung, mit mehreren umfangreichen technologischen Entwicklungen (im mechanischen, elektromechanischen und im elektronischen Bereich, im IT-Bereich, usw.).
- Es handelt sich dabei um einen industriellen Bereich mit hohen Sicherheits- und Verfügbarkeitsanforderungen; selbst in der Rückfallebene muss der Betrieb kontinuierlich aufrechterhalten werden (durch Vorschriften und Menschen).
- Das Bedienungspersonal muss die Anlagen technisch und funktional im Betrieb und bei der Instandhaltung im Griff haben.

## 1.2 Aufbau der Arbeit

Die vorliegende Arbeit umfasst die verschiedenen, zuvor beschriebenen Etappen. Eine Darstellung des Umfeldes bei der französischen Eisenbahn, die Studie der verschiedenen formalen Verfahren und die Einzelheiten der Vorgehensweise bei der Behandlung der Anwendungen finden sich im Anhang.

Im **Kapitel 2** wird zunächst die derzeitige Vorgehensweise bei der Realisierung und der Änderung kritischer IT-Systeme vorgestellt. Es wird auf die zunehmenden Schwierigkeiten bei der Validierung vor der Inbetriebnahme eingegangen. Diese Problematik findet sich auch außerhalb kritischer Eisenbahnsysteme wieder. Sie hat bereits einige wichtige und bekannte Unternehmen dazu gebracht, den Schritt hin zu kritischen IT-Systemen nicht zu vollziehen oder sogar eine frühere Entscheidung aus Gründen der mangelnden Beherrschbarkeit des Sicherheitsniveaus rückgängig zu machen.

**Kapitel 3** stellt die wichtigsten Ansätze zur Lösung dieser Problematik vor. Weiterhin werden die Konzeptionsrichtlinien für Rechner, die Architektur der Betriebssysteme und auch der Anwendersoftware behandelt. Generell handelt es sich bei der Vorgehensweise um die physische Realisierung eines endlichen Automaten.

D'une manière générale, la démarche présentée s'articule autour de la réalisation physique d'un automate à nombre fini d'états industriel et de l'utilisation des propriétés mathématiques de ce type d'entité pour réaliser la vérification exhaustive que le logiciel d'application est correct à la vue des exigences du système critique. Pour cela qu'il faut éviter l'algorithmique.

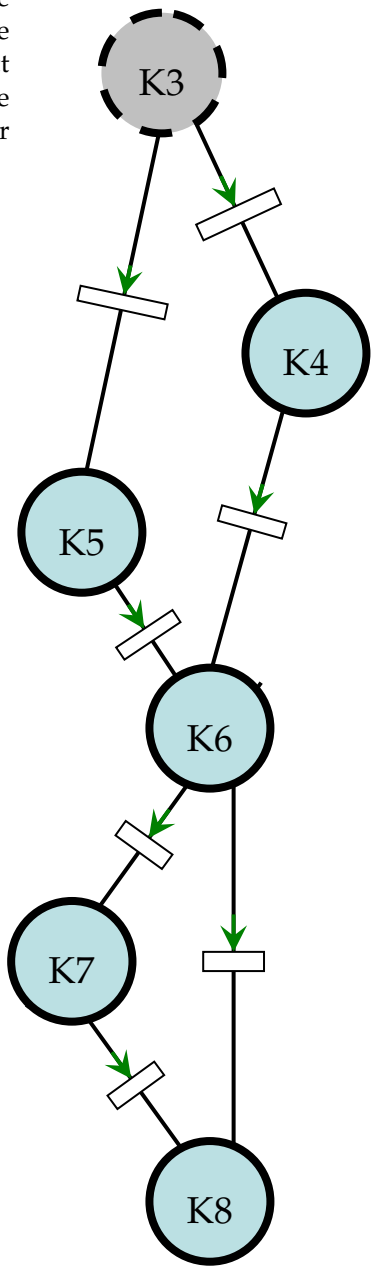
**Chapitre 4** : Nous décrirons le contexte ferroviaire afin d'en percevoir et formaliser les propriétés de sécurité logique et physique pour les systèmes informatiques critiques. D'une manière générale, il s'avère que la formalisation de ces propriétés requiert une compréhension large du système ferroviaire.

**Chapitre 5 :** Nous opèrerons un tour d'horizon des méthodes de nature à accroître le niveau de maîtrise de la sécurité des systèmes, des modèles de spécification semi formels et formels, de leur extension à la réalisation du code exécutable. Le chapitre 5 est chargé de présenter les points essentiels de la sécurité ferroviaire et les options françaises retenues pour couvrir les événements redoutés généraux identifiés

**Chapitre 6 :** Nous développerons la méthode formelle de notre choix reposant sur la théorie des automates et la réalisation physique d'un tel automate.

**Chapitre 7** : Nous appliquerons notre méthode à divers cas industriels, de la revalidation d'installations anciennes, de la validation d'installations contemporaines et de la spécification d'installations futures.

**Chapitre 8 :** Nous tirerons les conclusions et dégageons des perspectives pour l'avenir du développement des futurs systèmes informatiques critiques.




Die mathematischen Eigenschaften eines solchen Automaten werden benutzt, um eine erschöpfende Validierung durchzuführen, die gewährleistet, dass die Anwendersoftware den Anforderungen des sicherheitskritischen Systems entspricht.

**Kapitel 4** behandelt den Kontext der Eisenbahn. Daraus werden die logischen und physischen Eigenschaften eines sicherheitskritischen IT-Systems abgeleitet. Die Formalisierung dieser Eigenschaften benötigt eine breite Kenntnis des Eisenbahnsystems.

Das **Kapitel 5** gibt einen Überblick über die Verfahren, die dafür geeignet sind, die Sicherheit des Systems besser zu beherrschen. Semiformale und formale Spezifizierungsmodelle werden vorgestellt, sowie deren Erweiterungen bis hin zur Realisierung des ausführbaren Codes. In Kapitel 5 werden die wesentlichen Punkte der Bahnsicherheit aufgeführt, sowie die französischen Lösungen zur Vermeidung allgemein identifizierter befürchteter Ereignisse.

Im **Kapitel 6** wird die ausgewählte, auf der Theorie der Automaten basierende formale Methode entwickelt. Die physische Realisierung eines solchen Automaten wird vorgestellt.

Im **Kapitel 7** wird die ausgewählte Methode an verschiedenen industriellen Anwendungen erprobt: von der erneuten Validierung alter Anlagen und der Validierung aktueller Anlagen bis hin zu der Spezifikation zukünftiger Anlagen.

Im **Kapitel 8** werden Schlussfolgerungen gezogen und Perspektiven für zukünftige sicherheitskritische IT-Systeme aufgezeigt.

# CHAPITRE 2

# Problématique et objectif du travail

Comment faire des systèmes informatiques à haut niveau de sécurité à complexité croissante et tout en maîtrisant les coûts. C'est une problématique qui s'impose à tous les industriels tant concepteurs que gestionnaires de tels systèmes informatiques ou informatisés. L'expérience montre en effet que à mesure que les systèmes informatiques deviennent plus complexes les contraintes de coûts et de délais sont maintenues, leur mise au point est de plus en plus difficile et les tâches correspondant aux essais d'intégration et de validation sont de plus en plus lourdes.

Depuis une trentaine d'année, les systèmes informatisés ou informatiques réalisent de plus en plus des fonctions critiques, quelque soit le domaine industriel. Le retour d'expérience fait apparaître :

- une faible durée de vie (régénérations fréquente) ;
- des coûts et délais de développement et de tests importants ;
- une complexité de plus en plus croissante devant la concentration des traitements ;
- quelques incidents, voire accidents, malgré l'existence et l'application de normes européennes ou internationales dont les causes sont très majoritairement des erreurs de spécification, plutôt que des défauts matériels ;
- une séparation des métiers et des compétences tant informatiques que sûreté de fonctionnement ;
- une dépossession de la maîtrise technique système…

## 2.1 Préjugés persistants

Pour équilibrer l'équation, des pressions sont exercées sur les gestionnaires d'infrastructure et les autorités de sécurité pour que la variable d'ajustement soit le juste niveau de sécurité. Elles conduisent à l'installation de préjugés fréquemment entendus tels que:

- la sécurité coûte nécessairement cher ;
- les contraintes de sécurité sont les arguments des protectionnistes ;

# KAPITEL 2

# Problematik und Zielsetzung

Wie kann man immer komplexere IT-Systeme mit hohem Sicherheitslevel konzipieren und dabei auch noch die Kosten im Griff haben? Mit dieser Problematik sind viele Branchen konfrontiert: die Entwicklungsabteilungen, aber auch die Betreiber solcher IT-Systeme. Die Erfahrung zeigt in der Tat, dass, obwohl die IT-Systeme komplexer werden, der Druck auf die Kosten und die Fristen nicht nachlässt; die Fertigstellung solcher Systeme wird also immer schwieriger und die Aufgaben in Verbindung mit den Integrations- und Validierungsversuchen werden immer aufwendiger.

Seit ca. 30 Jahren führen informatisierte oder IT-Systeme immer mehr kritische Funktionen aus, in welchem industriellen Bereich auch immer. Aus den gesammelten Erfahrungen geht folgendes für diese Systeme hervor:

- sehr niedrige Lebensdauer (häufige Erneuerungen),
- hohe Entwicklungs- und Versuchskosten bzw. –zeiten,
- steigende Komplexität, aufgrund einer Konzentration der Verarbeitungsvorgänge,
- einige Vorfälle oder gar Unfälle, trotz Bestehen bzw. Anwendung europäischer oder internationaler Normen liegen die Ursachen überwiegend in Spezifikationsfehlern und weniger in Hardwarefehlern (Materialfehler),
- Diversifizierung der Berufe und der Fachkenntnisse, sowohl im IT-Bereich, also auch im Bereich der Funktionssicherheit,
- Verlust der technischen Beherrschung des Systems…

## 2.1 Beharrliche Vorurteile

Zur Bewältigung oben genannter Probleme wird auf die Infrastrukturbetreiber und die Sicherheitsbehörden Druck ausgeübt, damit das *richtige* Sicherheitsniveau gefunden wird. Dies führt zu den wohl bekannten und oft gehörten Vorurteilen, zum Beispiel:

- Sicherheit ist notwendigerweise teuer,
- Sicherheitsbedingungen sind Argumente der Protektionisten,

- l'assurance qualité au sens des normes actuelles est suffisante pour garantir le niveau de sécurité du logiciel fonctionnel applicatif ;
- la sécurité des systèmes informatiques est l'affaire d'experts en informatique.

Ceci alors même que des expériences infirmant ces propos apparaissent de manière de moins en moins marginale (voire même de manière quasi officielle récemment) [01Info, 2001] [Bombardier, 2005] [SNCF, 2005] [Lötschberg, 2007] [Lötschberg, 2008] [Gouda, 2008] [Hoch, 2008] [Gartner, 2008] [Gartner, 2009] [Barbier, 2009].

D'ailleurs, d'importants exploitants de systèmes industriels critiques[5], à la vue d'expériences passées (publiées ou non), ont pris l'option de ne plus utiliser de systèmes informatiques pour les fonctions de sécurité de plus haut niveau. Ceci essentiellement pour deux raisons, l'impossibilité de garantir que le fonctionnel applicatif est «100% correct» et la réduction de la disponibilité opérationnelle des fonctions.

## 2.2 Défis pour demain

Ces défis apparaissent de plus en plus clairement dans un contexte où :

- les installations informatiques industrielles sont de plus en plus complexes ;
- les coûts de développement croissent avec la taille des systèmes et la possibilité de centralisation ;
- les enjeux des projets de développement sont économiques avant même d'être sécuritaires ;
- le passage annoncé à des automatismes intégraux conduit à des exigences accrues, et une perte des connaissances métier ;
- les incidents ou les accidents peuvent avoir des conséquences médiatiques incalculables s'ils se répètent, l'exigence acceptable des populations est croissante ;
- la croissance du trafic rend les opérations de maintenance et de modification plus complexes pour ne pas réduire la capacité de production.

- Die Qualitätssicherung im Sinne der heutigen Normen genügt, um das Sicherheitsniveau der anwendungsbezogenen Software zu gewährleisten,
- Die Sicherheit der IT-Systeme ist Sache der IT-Experten.

Diese Vorurteile bleiben bestehen trotz der immer weniger marginalen Erfahrungen, die diese Aussagen widerlegen. Diese Feststellungen sind zum Teil sogar offiziell. [01Info, 2001] [Bombardier, 2005] [SNCF, 2005] [Lötschberg, 2007] [Lötschberg, 2008] [Gouda, 2008] [Hoch, 2008] [Gartner, 2008] [Gartner, 2009] [Barbier, 2009].

Der wirtschaftliche Einfluss scheint größer zu sein als die eventuellen menschlichen Kosten. Aufgrund von Erfahrungen (die veröffentlicht wurden oder auch nicht) haben wichtige Betreiber von kritischen industriellen Systemen[6] beschlossen, bei Sicherheitsfunktionen der höchsten Ebene auf IT-Systeme zu verzichten. Es ist nicht möglich zu garantieren, dass die Anwendungsfunktionen zu 100% korrekt sind und es kann sein, dass solche Systeme die betriebliche Verfügbarkeit der Funktionen beeinträchtigen.

## 2.2 Herausforderungen für die Zukunft

Die Herausforderungen für die Zukunft werden immer deutlicher in einem Umfeld in dem:

- die industriellen IT-Einrichtungen immer komplexer werden,
- die Entwicklungskosten mit der Größe der Systeme und mit der Zentralisierungsmöglichkeit steigen,
- bei den Entwicklungsprojekten die Wirtschaftlichkeit mehr als die Sicherheit zählt,
- der angekündigte Übergang zu integrierten Automatismen zusätzliche Anforderungen mit sich bringt, sowie den Verlust der Fachkenntnisse,
- wiederholte Zwischenfälle oder Unfälle unberechenbare Folgen in den Medien haben können und die Anforderungen seitens der Bevölkerung steigen,
- durch den Verkehrsanstieg Instandhaltung und Änderungen komplizierter werden, da die Produktionskapazitäten nicht reduziert werden sollen.

[5] C'est par exemple le cas de EDF (palier N4), l'armée française (missile nucléaire M51), SNCF (postes d'aiguillage de nouvelle génération PI 2006)…

[6] Dies ist zum Beispiel der Fall von EDF (Atomkraftwerk Typ „Palier N4"), der französischen Armee (Interkontinentalrakete M51), der SNCF (Stellwerke der neuen Generation PI 2006), etc.

La faiblesse endémique des systèmes informatiques repose sur le fait qu'au niveau du fonctionnel, le niveau de sécurité repose en fait uniquement sur l'assurance qualité mise en œuvre lors de son développement. Les contraintes imposées par les normes reposent en effet essentiellement sur l'architecture matérielle (approche probabiliste) et sur l'approche assurance qualité du logiciel. Il est à noter que la norme EN50128 [EN50128, 2001] cite les méthodes formelles comme un moyen à privilégier pour les équipements programmés de niveau SIL4.

Il s'avère particulièrement délicat de quantifier le niveau de sécurité de fonctions portées par un ensemble informatique, ce alors que l'occurrence d'accidents va coûter de plus en plus cher aux entreprises du fait des médias, de l'Internet et du niveau d'exigence des sociétés occidentales. Traiter la sécurité à la conception pourrait s'avérer plus économique sur le long terme, par opposition à des économies de développement qui conduisent à des systèmes moins sûrs et moins profitables.

## 2.3 Développements possibles

A mon sens, l'industrie devra progresser sur la voie de la rationalisation économique des tâches de spécification et de validation finale tout en améliorant le niveau de maîtrise des risques inhérents à ces nouveaux systèmes. Il faut trouver une autre voie qui permettrait de répondre aux contraintes économiques tout en préservant les niveaux de sécurité actuels.

Prenons exemple sur le passé qui a fait ses preuves : les installations les plus sûres, les plus fiables, les plus durables sont les plus simples quant elles sont conçues en adéquation avec leur environnement et leurs conditions d'usage. Cette piste est accessible si l'on revient aux fondements des fonctions à réaliser et non à des choix de réalisation technique.

Notons que les postes mécaniques et électro-mécaniques ont fait l'objet de longues années durant de validations formelles de leurs tables d'enclenchements [Plisson 1886] [Descubes 1898] depuis plus de 100 ans et bien avant que ces termes n'apparaissent avec l'informatique moderne.

Les méthodes formelles sont le prolongement naturel des méthodes conception par les modèles, d'abord semi formelles puis formelles. Afin de faire simple, économique et sûr, les méthodes formelles semblent maintenant incontournables. (Figure 2.1) Il reste aux industriels à trouver la meilleure manière de les appliquer afin d'agir efficacement sur le cycle de développement des systèmes informatiques critiques.

Aus meiner Sicht beruht bei IT-Systemen die inhärente Schwäche darauf, dass auf funktionaler Ebene das Sicherheitsniveau nur von dem bei der Entwicklung angewandten Grad der Qualitätssicherung abhängt. Die Anforderungen der Normen betreffen hauptsächlich die Hardwarearchitektur mit einem ganz geringen Softwareanteil [EN50128, 2001]. Es ist anzumerken, dass die EN50128 Norm formale Verfahren als ein bevorzugtes Mittel zur Überprüfung von SIL4 programmierter Software zitiert.

Die Quantifizierung des Sicherheitsniveaus von IT-Funktionen erweist sich als besonders heikel. Gleichzeitig werden, aufgrund der Medien, des Internets und der Anforderungen der westlichen Gesellschaft die Unfälle für die Betreiber immer teurer. Die Berücksichtigung der Sicherheit bei der Konzeption könnte sich längerfristig als günstiger erweisen verglichen mit Entwicklungseinsparungen, die zu weniger sicheren und weniger profitablen Systemen führen.

## 2.3 Mögliche Ansätze

Meines Erachtens muss die Industrie bei der wirtschaftlichen Rationalisierung der Spezifizierungs- und Validierungsaufgaben noch Fortschritte erzielen; sie muss gleichzeitig die mit diesen neuen Systemen einhergehenden Risiken noch besser beherrschen. Man muss daher einen neuen Weg finden, den wirtschaftlichen Erfordernissen Rechnung zu tragen und gleichzeitig die heutigen Sicherheitsniveaus aufrecht zu halten.

Es gibt Beispiele aus der Vergangenheit, die sich bewährt haben: die sichersten, zuverlässigsten und langlebigsten Anlagen sind dann am einfachsten, wenn sie unter Berücksichtigung des Umfeldes und der üblichen Bedingungen konzipiert werden. Dieser Weg ist dann möglich, wenn man sich auf die Grundlage der zu realisierenden Funktionen zurückbesinnt, und nicht nur die Wahl einer technischen Realisierung berücksichtigt.

Es ist anzumerken, dass die Verschlusstafeln der mechanischen und der elektromechanischen Stellwerke jahrelang formal geprüft wurden [Plisson 1886] [Descubes 1898] seit mehr als 100 Jahren und lange vor dem Auftreten dieser Begriffe im Zusammenhang mit der modernen Informatik.

Die formalen Methoden sind die natürliche Weiterentwicklung der Konzeptionsmethoden basierend auf Modellen, erst halbformalen, dann formalen. Um einfache, wirtschaftliche und sichere Systeme zu konzipieren, scheinen die formalen Verfahren unumgänglich zu sein (Abbildung 2.1). Die Industrie muss den besten Weg finden diese anzuwenden, um den Entwicklungszyklus kritischer IT-Systeme effizient zu beherrschen.

L'application de méthodes formelles repose sur l'expression complète par les experts métier des fonctionnalités attendues, des conditions d'usage et d'environnement, des propriétés de sécurité, le tout en lien avec les fondements du passé : un mal pour un bien en quelque sorte.

Die Anwendung formaler Verfahren beruht auf der vollständigen Beschreibung der erwarteten Funktionen, der üblichen Betriebsbedingungen, der Umweltbedingungen und der Sicherheitseigenschaften durch die Fachleute und dies in Verbindung mit den Grundlagen der Vergangenheit : dieser Aufwand muss leider akzeptiert werden.

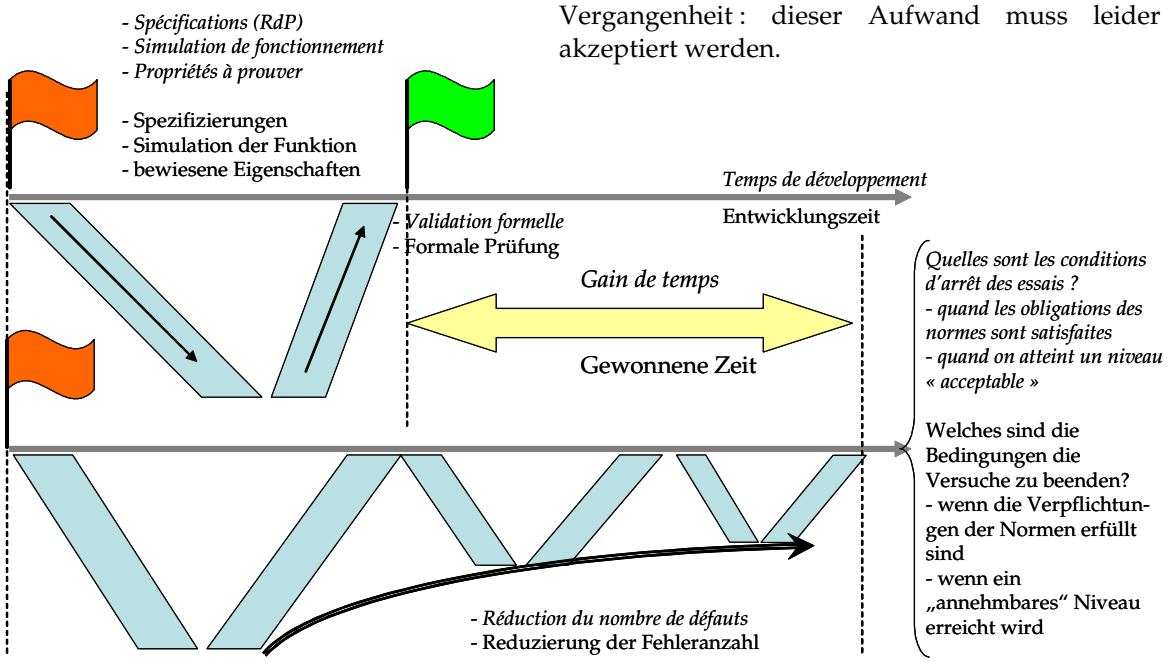



**Figure 2.1 :** Influence des méthodes formelles sur le cycle de développement
**Abbildung 2.1**: Einfluss formaler Methoden auf den Entwicklungszyklus

## 2.4 Objectifs du travail

Pour faire face à ces défis économiques et de sécurité, je proposerai une démarche de « conception – validation » applicables à des systèmes informatiques temps réels, critiques ou non.

Ce travail a pour objet d'explorer ce domaine des possibles, de proposer une méthode formelle permettant d'une part, une mise en oeuvre industrielle et, d'autre part, le maintien de la maîtrise fonctionnelle système par les experts signalisation.

Les méthodes formelles apparaissent prometteuses (au vu des cas d'applications industrielles en langage B ou avec iProover…) pour répondre à la fois aux enjeux économiques et sécurité [Monin, 1996]. Néanmoins aujourd'hui elles ne sont pas déployées massivement dans l'industrie.

Après un tour d'horizon des méthodes disponibles et des contraintes propres au domaine ferroviaire nous justifierons d'une part, le choix d'un langage de description interprétable du fonctionnel d'une application industrielle et, d'autre part, une méthode de validation formelle.

## 2.4 Zielsetzungen der Arbeit

Um sich diesen wirtschaftlichen und Sicherheitsherausforderungen stellen zu können, schlägt die vorliegende Arbeit eine „Konzeptions und Validierungs"-Methode vor, die auf kritische und nicht kritische Echtzeit-IT-Systeme anwendbar ist.

Das Ziel dieser Arbeit besteht darin, die Möglichkeiten zur industriellen Implementierung eines formalen Verfahrens zu erforschen und ein Verfahren zur Aufrechthaltung der funktionalen Beherrschung des Systems durch die Signaltechnikexperten vorzuschlagen.

Formale Verfahren scheinen (aufgrund der industriellen Anwendungen mit der Sprache B oder iProover, usw.) viel versprechend zu sein bei der Beantwortung der wirtschaftlichen und sicherheitstechnischen Herausforderungen [Monin, 1996]. Trotzdem werden sie zurzeit in der Industrie nicht massiv genützt.

Nach der Erörterung der schon bestehenden Verfahren und der Randbedingungen des Bahnsystems wird einerseits die Wahl der Sprache für die interpretierbare Beschreibung der Funktionen einer industriellen Anwendung und andererseits ein formales Validierungsverfahren beschrieben.

Ce travail vise à :

- minimiser les coûts de développement et de maintien en condition opérationnelle de systèmes informatiques critiques tels que les postes d'aiguillage ;
- proposer, dans des délais raisonnables, une analyse «exhaustive» des comportements possibles de systèmes informatiques de sécurité ;
- assurer la continuité entre le comportement prouvé du « modèle de spécification » et le comportement du « code embarqué » dans le système cible ;
- proposer une méthode de validation formelle reposant sur une description du fonctionnel de l'application utilisant une forme particulière des réseaux de Petri et permettant une interprétation déterministe de ce fonctionnel ;
- repousser les limites connues des techniques de transformation des modèles aux codes embarqués, en proposant une méthode qui permette de se passer de la transformation pour assurer la continuité entre le comportement prouvé du modèle et le comportement du code embarqué dans le système cible.

L'objectif est définir une méthode réellement applicable de manière industrielle minimisant les coûts et les délais de validation et garantissant son taux de couverture (cf. Figure 2.1).

Il s'agit de montrer que pour un domaine des possibles des entrées du système, celui-ci ne pourra atteindre un état non sûr au regard des conditions d'exploitation, des procédures de gestion des modes dégradés.

La méthode doit montrer que les fonctionnalités, dans un contexte donné, portées par un système informatique critique sont correctes (à 100%) et doit couvrir l'ensemble du cycle de développement du logiciel, des spécifications au code réellement installé dans le système cible.

La méthode doit pouvoir s'intégrer dans une problématique plus large de réalisation d'une étude de sûreté système, associant des éléments non informatiques et des éléments informatiques.

Ziel dieser Arbeit ist somit:

- die Minimierung der Entwicklungs- und Aufrechterhaltungskosten kritischer IT-Systeme (wie z.B. Stellwerke).
- der Vorschlag einer „vollständigen" Analyse aller möglicher Verhaltensweisen von IT-Sicherungssystemen die in einer annehmbaren Zeit durchführbar ist.
- die Gewährleistung der Kontinuität zwischen dem bewiesenen Verhalten des „Spezifizierungsmodells" und dem Verhalten des „Codes" des Zielsystems.
- der Vorschlag eines formalen Validierungsverfahrens auf der Grundlage der Beschreibung der Anwendungsfunktionen unter Verwendung einer besonderen Form von Petrinetzen, mit der Möglichkeit einer deterministischen Interpretation dieser Funktionen.
- die Aufhebung der bekannten Grenzen der Techniken zur Umwandlung der Modelle in Codes durch den Vorschlag eines neuen Verfahrens, das diese Umwandlung nicht benötigt um die Kontinuität zwischen dem bewiesenen Verhalten des „Spezifizierungsmodells" und dem Verhalten des „Codes" des Zielsystems zu sichern.

Das Ziel besteht darin, ein anwendbares Verfahren zu definieren, das die Validierungskosten und –fristen minimiert und den Deckungsgrad gewährleistet (vgl. Abb. 2.1).

Es geht darum, vollständig zu beweisen, dass bestimmte Eingänge das System nicht in einen unsicheren Zustand in Bezug auf die Betriebsbedingungen und die Vorschriften für das Management der Rückfallebenen bringen können.

Das Verfahren soll beweisen, dass die Funktionen eines kritischen IT-Systems 100%ig korrekt sind, und es soll den gesamten Softwareentwicklungszyklus abdecken, von den Spezifikationen bis hin zum Code der Zielmaschine.

Das Verfahren muss in eine weit gefächerte Problematik integriert werden, und zwar in eine Sicherheitsanalyse eines Systems, einschließlich der IT- und nicht-IT Elemente.

# CHAPITRE 3

# Orientations pour le travail

Le travail présenté dans ces lignes vise à apporter des réponses théoriques et pratiques à la problématique exposée plus avant. Les orientations prises s'appuient principalement sur les constats suivants.

## 3.1 Modifiabilité des systèmes informatiques

Les systèmes programmables ne sont pas si adaptables qu'on le croit. La dématérialisation ne signifie pas la simplification (au contraire bien souvent), ni la disparition du travail de conception, bien au contraire (la centralisation le rend encore plus complexe). L'adaptation d'un système mal conçu est coûteuse lorsqu'il faut reprendre le logiciel ou modifier le matériel. La raison en est simple : cela représente autant de travail. Les logiciels des systèmes programmables sont aussi rigides que leurs ancêtres mécaniques, la durée de vie en moins [SNCF, 2005] [Lévi, 1995].

Un système mal conçu, qu'il soit programmable ou non, est un système durablement mal conçu. Il nécessitera des rattrapages en exploitation.

Dans le même ordre d'idées, il ne faut pas s'attendre à ce que le recours à des technologies nouvelles résolve comme par miracle les difficultés fonctionnelles du système. Le risque est grand de passer outre les principes de sécurité qui ont fait leurs preuves depuis de nombreuses années.

La maintenance des installations de sécurité est non seulement avant tout préventive, mais repose aussi sur une représentation correcte de la philosophie du système. Certaines actions de maintenance préventive sont des interventions habituellement considérées comme correctives, parce qu'elles se font sur défauts latents et non sur défaillance.

# KAPITEL 3

# Arbeitsausrichtung

Die hier vorgestellte Arbeit beinhaltet einige theoretische und praktische Antworten auf die beschriebene Problematik. Die Ausrichtung der Arbeit stützt sich vor allem auf die nachstehend geschilderten Feststellungen.

## 3.1 Veränderungsmöglichkeiten bei IT-Systemen

Programmierbare Systeme lassen sich nicht so leicht anpassen, wie man meint. Dematerialisierung ist nicht gleich Vereinfachung (es ist sogar oft genau das Gegenteil) und lässt zusätzlich nicht die Konzeptionsarbeit verschwinden. Ganz im Gegenteil: die Zentralisierung macht die Entwicklung noch schwieriger. Die Anpassung eines schlecht konzipierten Systems ist kostspielig, wenn die Soft- oder Hardware geändert werden muss, da dies noch einmal die gleiche Arbeit wie bei der Konzeption beinhaltet. Die Software programmierbarer Systeme ist genauso unflexibel wie die mechanischen Vorgänger, jedoch ohne deren Lebensdauer[SNCF, 2005] [Lévi, 1995].

Ein schlecht konzipiertes System, sei es programmierbar oder nicht, ist langfristig schlecht konzipiert. Es muss während des Betriebs verbessert werden.

Genauso darf man nicht davon ausgehen, dass neue Technologien Wunder bewirken und die funktionalen Schwierigkeiten des Systems beheben. Es besteht in diesem Fall nämlich ein großes Risiko, sich über Sicherheitsprinzipien hinwegzusetzen, die sich schon jahrelang bewährt haben.

Die Instandhaltung der Sicherheitsanlagen hat nicht nur einen vorbeugenden Charakter, sondern beruht auch auf einer korrekten Darstellung der Systemphilosophie. Bestimmte Eingriffe, die im Rahmen der vorbeugenden Instandhaltung durchgeführt werden, gelten üblicherweise als korrektive Eingriffe, da sie latente Fehler und nicht Ausfälle betreffen.

Ces actions sont préventives dans le sens où elles doivent se faire même si les défauts n'ont pas de conséquence. Tout autant que réduire la probabilité d'apparition d'un second défaut, cette prévention a une fonction non dite, qui est d'éviter que ne s'instaure un « délit d'habitude ». En effet, il y a risque à laisser s'instaurer un bon fonctionnement habituel avec des défauts persistants « sans gravité ». Quoi qu'il en soit, le développement des automatismes (substitution des objets techniques à l'homme) tend à escamoter temporairement les risques et à les faire oublier.

## 3.2 Problématique de la sécurité des systèmes informatiques critiques

La sûreté de fonctionnement peut être définie [Villemeur, 1997] [Pages, 1980] comme une confiance justifiée dans la capacité d'un système à réaliser correctement les fonctions attendues. La sûreté quant à elle peut être définie comme la capacité démontrable d'un système à éviter ou à faire face aux situations dangereuses et inacceptables. Le terme «système» est ici utilisé dans un sens large. A un bout de l'échelle, il peut désigner une installation industrielle complète comme une tranche nucléaire, un poste d'aiguillage, un réseau de transport électrique ou un système de transport. Il peut aussi désigner un composant majeur d'une installation, comme un système de contrôle - commande ou un équipement électromécanique. Enfin, à l'autre bout de l'échelle, le terme «système» peut désigner un composant élémentaire comme un circuit intégré ou une carte électronique.

### 3.2.1 Liens avec le passé

Le domaine ferroviaire a ceci de particulier qu'il n'existe pas d'Analyse Préliminaire des Risques Système (APRS) du système ferroviaire, ou alors de très haut niveau, les compétences se transfèrent par compagnonnage. De ce fait, les explications des raisons de certains choix passés peuvent sembler flous aux non initiés et, si certaines simplifications sont possibles compte tenu des évolutions des technologies, certaines autres peuvent se révéler rapidement dangereuses. Les principes de signalisation élaborés et affinés au cours du temps capitalisent toutes les expériences acquises depuis plus d'une centaine d'année.

Es sind insofern vorbeugende Eingriffe, als dass sie notwendig sind, selbst wenn die Fehler keine Konsequenzen haben. Dieser vorbeugende Eingriff reduziert die Wahrscheinlichkeit eines zweiten Fehlers und verhindert „Gewohnheitsdelikte". Es besteht ein Risiko, dass sich eine gewohnt korrekte Funktionsweise mit langlebigen „harmlosen" Fehlern einpendelt. Die Entwicklung von Automatismen (Ersatz des Menschen durch die Technik) führt zu einem vorübergehenden Verschwinden der Risiken, die dann vergessen werden.

## 3.2 Sicherheitsproblematik kritischer Rechnersysteme

Die Funktionssicherheit wird hier [Villemeur, 1997] [Pages, 1980] definiert als das gerechtfertigte Vertrauen in die Fähigkeit eines Systems, die erwarteten Funktionen korrekt auszuführen. Die Betriebssicherheit, hingegen, wird definiert als die nachweisbare Fähigkeit eines Systems, gefährliche bzw. unakzeptable Situationen zu vermeiden oder zu bewältigen. Der Begriff System wird hier im weitesten Sinne benützt. Auf der einen Seite bezeichnet er eine vollständige industrielle Anlage, z.B. einen nuklearen Reaktor, ein Stellwerk, ein elektrisches Transportnetz oder ein Transportsystem. Es bezeichnet auch einen wesentlichen Bestandteil einer Anlage, z.B. ein Zugsteuerungssystem oder eine elektromechanische Einrichtung. Auf der anderen Seite kann es ein elementarer Bestandteil, z.B. eine elektronische integrierte Schaltung oder eine elektronische Steckkarte bezeichnen.

### 3.2.1 Verbindungen zur Vergangenheit

Der Bahnbereich ist insofern ein besonderer Bereich, da es keine vorläufige Risikoanalyse des Systems gibt und wenn dann nur auf einem sehr hohen Niveau; die Kompetenzen werden unter Kollegen weitergegeben. Deswegen sind die Gründe bestimmter, früherer Entscheidungen für Laien unklar. Selbst wenn bestimmte Vereinfachungen aufgrund technologischer Entwicklungen möglich sind, so erweisen sich manche aus diesem Grund sehr schnell als gefährlich. Die Grundsätze des Signalsystems, die im Laufe der Zeit aufgestellt und verfeinert wurden, beinhalten die Erfahrungen aus über mehr als hundert Jahren.

Ainsi, parle-t-on encore aujourd'hui de l'enclenchement de LAGNY, mis en place suite à l'accident survenu en 30 décembre 1933 où un train rattrapa en vitesse un train à l'arrêt faisant 200 morts et 300 blessés. Ou encore de celui de GAGNY.

## 3.2.2 Problématiques particulières des systèmes critiques

Les systèmes technologiques présentant des enjeux importants en matière de sûreté de fonctionnement sont aujourd'hui conçus, fabriqués et exploités de manières variables selon les secteurs d'activité (nucléaire, aéronautique, ferroviaire, automobile, industries), mais restent fondamentalement une combinaison de quelques principes comme par exemple :

- Redondance : les éléments critiques sont multipliés, de manière à ce qu'aucune défaillance aléatoire d'un seul élément ne puisse causer la défaillance du système ;
- Défense en profondeur : un chemin critique accidentel doit être couvert par plusieurs barrières successives et indépendantes ;
- Diversification : deux voies redondantes doivent employer des technologies différentes ;
- Ségrégation : les anomalies doivent être détectées et confinées, de manière à éviter qu'une erreur puisse se propager à l'ensemble du système et causer sa défaillance ;
- Prédictibilité : les mêmes causes doivent produire les mêmes effets, de manière à ce que le fonctionnement soit déterministe ;
- Robustification : le système doit résister à des agressions environnementales exceptionnelles.

L'application de ces principes, adaptés à la technologie informatique, aux enjeux de sûreté de fonctionnement, aux contraintes économiques et réglementaires, à l'histoire et à l'expérience de chaque secteur d'activité, forment un « état de l'art », permettant de concevoir, fabriquer et exploiter des systèmes sûrs, c'est à dire ayant un niveau de risque toléré par la société [Barbier, 2009].

Deswegen spricht man z.B. heute noch vom Verschluss von Lagny, der nach dem Unfall am 30.12.1933 eingesetzt wurde: damals hat ein Zug einen stehenden Zug eingeholt und es gab 200 Tote und 300 Verletzte. Ferner ist in diesem Zusammenhang der Unfall von Gagny zu erwähnen (vgl. Kap 5).

## 3.2.2 Problematik sicherheitsrelevanter Systeme

Die im Hinblick auf Betriebssicherheit mit wichtigen Herausforderungen behafteten technologischen Systeme werden heute je nach Branche (Kernenergie, Eisenbahn-, Auto-, Luftfahrtindustrie) unterschiedlich entwickelt, gefertigt und betrieben, aber sie bleiben grundsätzlich eine Kombination einiger Prinzipien, wie:

- der Redundanz: kritische Elemente sind mehrmals vorhanden, damit keine zufallsbedingte Störung eines einzigen Elementes eine Systemstörung bewirken kann.
- der gestaffelten Abwehr: ein durch einen Unfall herbeigeführter, kritischer Pfad wird durch mehrere unabhängige Schranken geschützt.
- der Diversifizierung: zwei redundante Pfade beruhen auf unterschiedlichen Technologien.
- der Trennung: Anomalien müssen festgestellt werden, um zu verhindern, dass ein Fehler Rückwirkungen auf das gesamte System hat, und eine Systemstörung bewirkt.
- der Vorhersehbarkeit: dieselben Ursachen führen zu denselben Folgen, im Sinne einer deterministischen Funktionsweise.
- der Robustheit: das System muss außergewöhnlichen Umweltbedingungen standhalten.

Die Anwendung dieser Prinzipien auf die Betriebssicherheit, auf wirtschaftliche und rechtlichen Randbedingungen, auf die Geschichte und auf die Erfahrung jedes Berufsfeldes, nach einer Anpassung an den IT-Bereich, führt zu einem Stand der Technik, der es erlaubt, sichere Systeme zu planen, herzustellen und zu nutzen, d.h. Systeme mit einem von der Gesellschaft akzeptierten Risiko [Barbier, 2009].

Du fait de cet état de l'art, la plus grande partie du risque des systèmes sûrs provient maintenant non plus de défaillances aléatoires, mais de «défaillances systémiques » [Leveson, 2000] [Leveson, 2001] [Bied, 2003] [Staffelbach, 2008], affectant l'ensemble d'un système, comme par exemple les défaillances de cause commune impactant simultanément tous les matériels d'un système redondé, les défaillances systématiques des logiciels dues à des erreurs de conception, les défaillances liées par des effets dominos, les défaillances dues à la malveillance ou à des catastrophes naturelles affectant simultanément des évènements initiateurs et des barrières.

Le risque systémique peut alors être défini comme la potentialité de défaillances systémiques. Des approches systémiques ont été ou sont mises en place pour maîtriser la complexité, en particulier par la création de modèles globaux d'installation permettant d'étudier les propriétés importantes telles que la sûreté ou la disponibilité. Ces modèles FMDS quantitatifs, basés sur des approches probabilistes, sont assez bien acceptés pour leur représentation des éléments purement matériels (défaillances aléatoires) et des facteurs humains. Ils le sont beaucoup moins pour leur représentation des aspects déterministes des systèmes informatiques (défaillances systémiques ou systématiques) [Leveson, 2000] [Leveson, 2001] [Bied, 2003] [Gartner, 2008] [Gartner, 2009] [SNCF, 2005] [Bombardier, 2005] [Lötschberg, 2007] [Lötschberg, 2008].

Il est de plus en plus difficile de trouver sur le marché des équipements techniques non programmés. Cette tendance est générale dans pratiquement toute l'industrie et les marchés grand public. Cette omniprésence se justifie par les bénéfices apportés par les techniques programmées. En particulier, elles permettent de réaliser des fonctions avancées non réalisables par des techniques plus conventionnelles. Des fonctions comme l'auto surveillance, l'auto étalonnage, la surveillance en ligne, la commande fine des équipements électromécaniques, l'installation de capteurs sans câblage supplémentaire peuvent améliorer significativement la fiabilité, la disponibilité et la sûreté. De plus, l'intégration toujours plus poussée des composants électroniques réduit de façon très significative le nombre de composants élémentaires et de pièces mobiles, réduisant ainsi le risque de pannes du matériel.

Aufgrund dieses Stands der Technik ist der größte Teil des Risikos bei sicheren Systemen nun nicht mehr auf zufallbedingte Störungen zurückzuführen, sondern auf „systembedingte Störungen" [Leveson, 2000] [Leveson, 2001] [Bied, 2003] [Staffelbach, 2008], die das ganze System beeinträchtigen. Als Beispiel sind hier aufzuführen: Fehler mit gemeinsamer Ursache, die gleichzeitig die ganze Hardware eines redundanten Systems beeinträchtigen, systematische Softwarefehler aufgrund von Konzeptionsfehlern, durch Ketteneffekte verbundene Fehler, Fehler aufgrund von Mutwilligkeit oder von Naturkatastrophen, die gleichzeitig die auslösenden Ereignisse und die Schranken beeinflussen.

Der Risikolevel eines Systems kann somit als ein möglicher systemischer Fehler definiert werden. Systemische Ansätze werden eingesetzt um die Komplexität zu beherrschen, insbesondere durch die Schaffung von Gesamtmodellen der Anlage, mit denen wichtige Eigenschaften (Betriebssicherheit oder Verfügbarkeit) untersucht werden können. Diese auf Wahrscheinlichkeitskonzepten beruhenden quantitativen RAMS-Modelle werden für die Darstellung rein materieller Elemente (zufallsbedingte Fehler) und des Faktors Mensch weitgehend akzeptiert. Sie werden weniger akzeptiert wenn es sich um die Darstellung deterministischer Aspekte von Rechnersystemen (systemische bzw. systematische Fehler) handelt [Leveson, 2000] [Leveson, 2001] [Bied, 2003] [Gartner, 2008] [Gartner, 2009] [SNCF, 2005] [Bombardier, 2005] [Lötschberg, 2007] [Lötschberg, 2008].

Es wird immer schwieriger, auf dem Markt nicht programmierbare technische Geräte zu finden: dies ist fast in der gesamte Industrie und auf den Verbrauchermärkten ein allgemeiner Trend. Diese Tatsache wird durch die Vorteile der programmierbaren technischen Geräte gerechtfertigt. Dadurch können insbesondere fortgeschrittene Funktionen realisiert werden, die eine herkömmlichere Technik nicht ermöglicht. Funktionen wie beispielsweise die Selbstüberwachung, die Selbstvalidierung, die Onlineüberwachung, die Feinsteuerung der elektromechanischen Teile oder der Einbau von Sensoren ohne zusätzliche Verkabelung, können die Zuverlässigkeit, die Verfügbarkeit und die Betriebssicherheit deutlich verbessern. Es kommt hinzu, dass die immer größere Integration der elektronischen Komponenten die Anzahl der elementaren Bestandteile und der beweglichen Teile sowie das Störungsrisiko der Hardware deutlich reduziert.

Cependant, les techniques programmées n'ont pas que des avantages. Outre une durée de vie commerciale courte et un vieillissement rapide [Antoni, 2008-1] [Iwata, 2008] de beaucoup d'équipements programmés, les principales difficultés viennent de l'introduction de nouveaux modes de défaillance et de vieillissement, et de l'augmentation importante de la complexité. Une plus grande complexité conduit à son tour à un risque plus élevé d'erreurs de spécification, de conception, et à une plus grande difficulté à démontrer les composantes de la SdF.

En fait, dans les équipements programmés, il est en pratique impossible de garantir l'absence de défauts de spécification et de conception, notamment dans le logiciel. Dans les équipements programmés de très haute qualité et de très haute fiabilité, ces erreurs résiduelles correspondent généralement à ce à quoi on n'a pas pensé. Il est donc extrêmement difficile, voire impossible, à déterminer la nature, l'impact et le nombre des erreurs résiduelles.

Les techniques numériques ayant un caractère fortement déterministe et répétable (le même comportement sera obtenu chaque fois que l'équipement sera mis dans les mêmes conditions), ces erreurs de conception résiduelles peuvent faire courir un risque de défaillance de cause commune, où plusieurs équipements partageant la même erreur et soumis à des conditions identiques tombent en panne de façon concomitante [Bied, 2003] [Nguyen, 2008].

**Contrairement aux défaillances dues aux défauts matériels qui apparaissent de façon aléatoire (hors mode commun), les méfaits des erreurs de conception résiduelles ne peuvent pas toujours être prévenus par la redondance. En effet, une erreur de spécification « non sûre » pouvant être « réveillée » par une combinaison particulière des entrées affecte alors irrémédiablement les sorties quelque soit le nombre d'unités en redondance, comme le montre la figure 3.1.**

Programmierbare Geräte haben jedoch nicht nur Vorteile. Die kommerzielle Lebensdauer ist kurz und zahlreiche programmierte Anlagen altern schnell [Antoni, 2008-1] [Iwata, 2008]. Die Hauptschwierigkeit besteht in der Einführung neuer Fehler- und Alterungsmodi, sowie der immer größeren Komplexität, die ebenfalls ein größeres Risiko von Spezifikations- und Konzeptionsfehlern birgt; dadurch wird der Nachweis der Betriebssicherheit schwieriger.

Eigentlich ist es in der Praxis unmöglich, bei programmierbaren Geräten zu gewährleisten, dass es keine Spezifikations- bzw. Konzeptionsfehler gibt, insbesondere bei der Software. Bei programmierbaren Geräten mit sehr hoher Qualität und sehr hoher Zuverlässigkeit entsprechen diese Restfehler im Allgemeinen den Punkten, die vergessen wurden. Es ist also extrem schwierig, wenn nicht unmöglich, die Art, den Einfluss und die Anzahl der Restfehler zu bestimmen.

Da die digitale Technik einen stark deterministischen und wiederholbaren Charakter aufweist (jedes Mal wenn das Gerät unter denselben Bedingungen funktioniert, ergibt sich dasselbe Verhalten), besteht - aufgrund dieser Restkonzeptionsfehler - das Risiko eines Ausfalls gemeinsamer Ursache: mehrere Geräte unterliegen demselben Ausfallmechanismus und sind gleichzeitig gestört, wenn sie identischen Bedingungen ausgesetzt werden [Bied, 2003] [Nguyen, 2008].

**Im Gegensatz zu Störungen aufgrund von Hardwarefehlern, die zufallsbedingt sind, kann die Redundanz die Auswirkungen der Restkonzeptionsfehler nicht immer vermeiden. Es ist in der Tat so, dass ein „unsicherer" Spezifikationsfehler, der durch eine spezielle Kombination der Eingänge „erwacht", unwiderruflich die Ausgänge beeinflusst und zwar unabhängig von der Anzahl der redundanten Einheiten, wie es Abbildung 3.1 zeigt.**

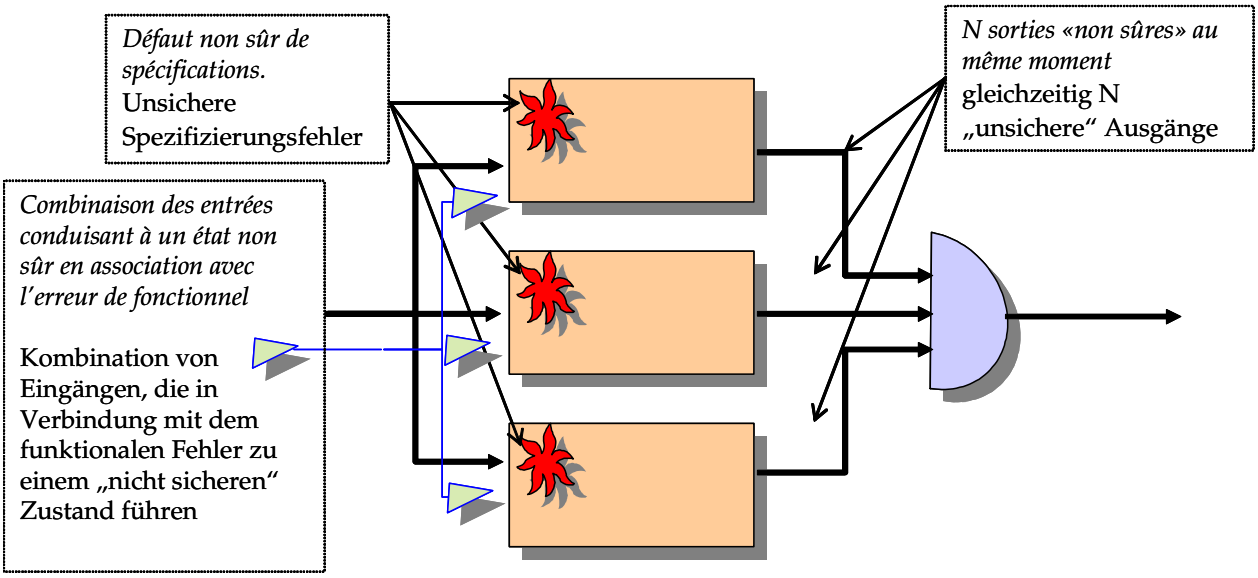



**Figure 3.1 :** Effet d'une erreur de spécification sur une architecture en N parmi P

**Abbildung°3.1**: Auswirkung eines Spezifizierungsfehlers auf eine N aus P Architektur

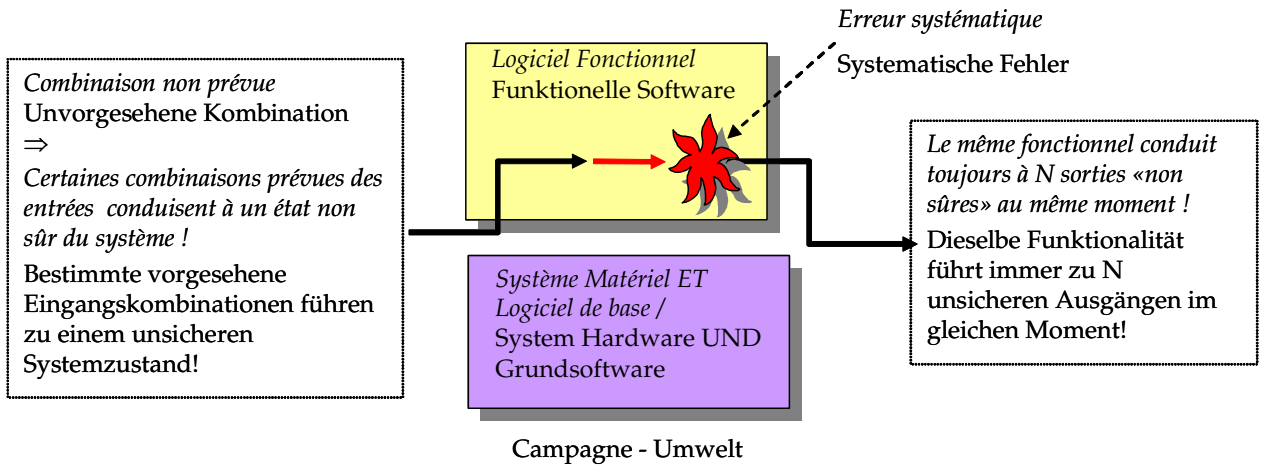



**Figure 3.2** : Les défauts de logiciels applicatifs conduisent de manière déterministe à une sortie non sûre du système informatique, quel que soit le niveau de fiabilité de l'ensemble matériel logiciel support

**Abbildung 3.2:** Anwendungssoftwarefehler führen deterministisch zu einem nicht sicheren Rechnerausgabewert, unabhängig vom Zuverlässigkeitsniveau der Kombination Hardware-Trägersoftware

## 3.3 Probabilité et sécurité informatique

La fiabilité opérationnelle et la sécurité des systèmes informatiques ou informatisés ne peuvent être estimées par des approches probabilistes, du moins dès lors que le niveau visé est industriel. Il est communément [CALIFE, 2001] [01Info, 2001] [01Info, 2002] [Chartier, 2003] [SNCF, 2005] admis que les approches dites à croissance de fiabilité ne sont d'aucun secours. Il n'est pas raisonnable de donner, par analogie avec les approches matérielles, un taux de fiabilité à un composant logiciel (regroupant d'ailleurs sans distinction les logiciels systèmes et ceux fonctionnel applicatifs).

La fiabilité opérationnelle et la sécurité des systèmes informatiques doivent, dans un contexte environnemental et d'usage donné, être considérées comme déterministes sur le plan des fonctionnalités. En effet, s'il vient à exister un «trou d'enclenchement», un cas non couvert par les spécifications fonctionnelles, chaque fois que la configuration non envisagée des entrées va se produire, le fonctionnel va conduire de manière «déterministe» à un état non sûr du système, quelque soit par ailleurs son architecture matérielle et son niveau de sécurité au sens des normes actuelles (SIL) [Nguyen, 2008].

De plus, dans certains cas, il suffit que cette combinaison redoutée se produise une fois (Figure 3.2) pour que l'état atteint et mémorisé soit tel que le fonctionnel présente alors de multiples faiblesses dangereuses [Bombardier, 2005] [SNCF, 2005] [Gartner, 2009] [Lötschberg, 2007] [Lötschberg, 2008] [Gouda, 2008].

Les générateurs de tests sont des outils qui permettent de générer des séquences de tests qui ont la particularité d'être d'une part conformes aux spécifications ; d'autre part conformes à des objectifs de tests spécifiques. Ils permettent ainsi de contrôler que le système réel est conforme au cahier des charges initial. Cette technique est très bien adaptée pour les systèmes répartis et notamment pour les protocoles. Force est de constater que pour une application logicielle, la mise en oeuvre d'un plan de test ne s'apparente en aucun cas à l'application d'une méthode formelle, qu'il ne donne aucune garantie quant à l'exhaustivité des tests effectués.

## 3.3 Wahrscheinlichkeit und Sicherheit der IT-Systeme

Die betriebliche Zuverlässigkeit und die Sicherheit der Rechnersysteme bzw. der computergesteuerten Systeme können nicht mit Wahrscheinlichkeitskonzepten bewertet werden, zumindest dann nicht, wenn der angestrebte Sicherheitslevel für ein industrielles System gültig sein soll. Im Allgemeinen [CALIFE, 2001] [01Info, 2001] [01Info, 2002] [Chartier, 2003] [SNCF, 2005] wird davon ausgegangen, dass die so genannten Ansätze der steigenden Softwarezuverlässigkeit auch nicht hilfreich sind; denn im Gegensatz zu physischem Material kann man bei einer Softwarekomponente (unabhängig davon, ob es sich dabei um eine System- oder um eine Anwendungssoftware handelt) keine statistische Zuverlässigkeitsrate berechnen. In einem gegebenen Umfeld und für eine bestimmte Verwendung müssen die betriebliche Zuverlässigkeit und die Funktionssicherheit der Rechnersysteme als deterministisch angesehen werden. Falls es in der Tat zu einem „Sicherungsloch" - einem von den funktionellen Spezifikationen nicht abgedeckten Fall - kommt, wird (jedes Mal, wenn die nicht vorgesehene Eingangskonfiguration auftritt) die „deterministische" Funktionsweise zu einem unsicheren Systemzustand führen und zwar unabhängig von der Materialarchitektur und dem Sicherheitsniveau im Sinne der heutigen Normen. [Nguyen, 2008]

In einigen Fällen reicht es aus, dass die befürchtete Kombination einmal (Abb. 3.2) vorkommt, damit der erreichte und gespeicherte Zustand dazu führt, dass die Funktionsweise zahlreiche gefährliche Schwächen aufweist [Bombardier, 2005] [SNCF, 2005] [Gartner, 2009] [Lötschberg, 2007] [Lötschberg, 2008] [Golda, 2008].

Testfallgeneratoren sind Programme zur Erzeugung von Testfallfolgen mit folgender Besonderheit: sie entsprechen sowohl den Lieferbedingungen, als auch den Zielsetzungen spezifischer Testfälle. Dadurch kann man überprüfen, ob das vorliegende System dem ursprünglichen Lastenheft mehr oder weniger entspricht. Diese Technik eignet sich sehr gut für verteilte Systeme, insbesondere für Protokolle. Es ist anzumerken, dass die Implementierung eines Testfalls bei einer Softwareanwendung keinesfalls der Anwendung eines formalen Verfahrens ähnelt; es gibt dabei nämlich keine Garantie der Vollständigkeit der Testfälle.

Le taux de couverture est nécessairement inférieur à l'unité compte tenu [Descubes, 1898] [Laprie, 1984] [Horstmann, 2005] [SNCF, 2005] [Staffelbach, 2008] :

- de l'impossibilité temporelle (voire technique) de réaliser l'ensemble des situations prévues (de par les contraintes temporelles notamment) ;
- de la difficulté de penser à bâtir des tests pour vérifier le respect de fonctionnalités (ou de domaines des entrées) non considérées lors des études de spécification (peut-on penser à tester à ce que l'on à oublié précédemment ?).

Une génération de tests suivie d'une campagne de réalisation de ces tests pour l'application logicielle étudiée, devra avoir fait l'objet d'un processus de qualification formelle de la génération de ces tests, par exemple, voir des points de vue suivants :

- couverture des chaînes fonctionnelles ciblées ;
- périmètre de variation des données et variables d'entrées du ou des programme(s) concerné(s) ;
- protection du système vis-à-vis d'un référentiel de situations redoutées dans des environnements bien définis.

**Fort de ce constat, la seule voie disponible, même si elle est communément fortement décriée, réside dans l'application industrielle d'une méthode formelle. Les recherches opérées** [Bielinski, 1993] [Narboni, 2001] [Schlingoff, 2002] [Esposito, 2007] **montrent que ces méthodes bien connues dans les milieux universitaires ne sont que rarement ou partiellement mises en œuvre dans le milieu industriel.**

# 3.4 Approches probabilistes et déterministes des systèmes

## 3.4.1 Approches déterministes

Dans ce type d'approche, on cherche à comprendre les caractéristiques et le comportement des principaux composants du système, et à identifier les mécanismes et les modes de défaillance.

Cette voie, souvent dite «déterministe», couvre les aspects matériels (composants mécaniques, composants électroniques…), susceptibles de conduire à des défauts apparaissant de façon aléatoire du fait par exemple du vieillissement, et les défauts de conception, notamment dans le logiciel et l'architecture système, susceptibles de conduire à des défaillances « systématiques » se produisant chaque fois que l'équipement programmé est mis dans les mêmes conditions.

Der Abdeckungsgrad ist aus folgenden Gründen auf jeden Fall kleiner [Descubes, 1898] [Laprie, 1984] [Horstmann, 2005] [SNCF, 2005] [Staffelbach, 2008]:

- zeitliche (oder gar technische) Unmöglichkeit, alle vorgesehenen Situationen zu realisieren
- Schwierigkeit Testfälle zur Überprüfung der Einhaltung der Funktionalitäten zu erstellen (bzw. der Eingabebereiche), die bei der Untersuchung des Designs nicht berücksichtigt wurden (Ist es überhaupt möglich, an Testfälle zu denken, die zuvor vergessen wurden?).

Die Testfallerzeugung und die Anwendung der Testfälle bedarf eines formalen Verfahrens zur Qualifizierung der Testfallerzeugung vor allem im Hinblick auf:

- die Abdeckung gezielter funktioneller Ketten
- die Spanne der Änderungen der Eingangs-daten und Variablen des betreffenden Programms
- den Schutz des Systems vor einer Reihe von befürchteten Situationen in einem ganz bestimmten Umfeld.

**Aufgrund dieser Feststellung ist die industrielle Anwendung eines formalen Verfahrens - auch wenn sie im Allgemeinen stark angefochten wird - der einzig mögliche Weg. Aus Forschungsarbeiten** [Bielinski, 1993] [Narboni, 2001] [Schlingloff, 2002] [Esposito, 2007] **geht hervor, dass diese, innerhalb der Universitäten wohl bekannten Verfahren, von der Industrie nur selten oder nur teilweise umgesetzt werden.**

# 3.4 Stochastische und deterministische Ansätze für IT-Systeme

## 3.4.1 Deterministischer Ansatz

Bei dieser Art von Ansatz versucht man, die Eigenschaften und das Verhalten der Hauptbestandteile des Systems zu begreifen und die Störungsmechanismen und die Versagensarten zu identifizieren.

Dieser oft als „deterministisch" bezeichnete Ansatz umfasst die Hardwareaspekte, die zu zufälligen Fehlern (beispielsweise aufgrund von Alterung) und zu Konzeptionsfehlern führen können, insbesondere die Software und die System-architektur, die zu „systematischen" Störungen führen können, und zwar jedes Mal wenn die programmierte Einrichtung unter denselben Bedingungen funktioniert.

## 3.4.2 Approches probabilistes

Les approches systémiques cherchent à représenter dans un modèle probabiliste global les différents mécanismes susceptibles de conduire à des défaillances ou accidents graves dans une installation.

De ce fait, elles renoncent à étudier les détails des mécanismes pour n'en garder qu'une vision macroscopique. Traditionnellement, elles ne représentaient **que les pannes aléatoires du matériel**.

Il est cependant apparu nécessaire de tenir compte de deux autres types de phénomènes:

- Les erreurs dans les interactions Homme - Machine notamment dans l'application des dispositions règlementaires.
  ⇒ défauts stochastiques ;
- Les défaillances systématiques causées par les défauts de spécification, de conception…
  ⇒ défauts déterministes.

## 3.4.3 Complémentarité des approches

Jusqu'à aujourd'hui, les approches probabiliste et déterministe s'opposaient quant à l'évaluation du niveau de sécurité et de disponibilité d'un système informatisé. Ceci est vrai dans toutes les organisations (exploitants, autorités de sûreté, fournisseurs) et dans tous les pays concernés par le ferroviaire et également dans toutes les industries concernées par la sûreté.

Les systèmes industriels complexes associent à la fois des fonctions physiques et des fonctions informatisées. Il serait utile de définir une méthode générale pour évaluer l'impact des équipements programmés dans les études probabilistes de sûreté combinant les deux approches.

Pour donner une réponse adéquate à cette question, des verrous doivent encore être levés.

En particulier :

## 3.4.2 Stochastischer Ansatz

Beim stochastischen Ansatz versucht man in einem globalen Wahrscheinlichkeitsmodell die verschiedenen Mechanismen darzustellen, die zu schwerwiegenden Fehlern oder Unfällen einer Anlage führen können.

Es wird dabei auf die Untersuchung der Einzelheiten der Mechanismen verzichtet: man betrachtet das System lediglich makroskopisch. Traditionell werden nur die **zufälligen Störungen der Hardware** dargestellt. Es scheint jedoch notwendig, auch zwei andere Erscheinungen zu berücksichtigen:

- Fehler in den Wechselwirkungen Mensch - Maschine, insbesondere bei der Anwendung der Bestimmungen der Regelwerke
  ⇒ stochastische Fehler
- Systematische Fehler aufgrund von Spezifikationsfehlern, Konzeptionsfehlern...
  ⇒ Deterministische Fehler.

## 3.4.3 Komplementarität der Konzepte

Bis heute stehen sich die auf Wahrscheinlichkeit beruhenden und die deterministischen Konzepte bei der Bewertung des Sicherheitslevels und der Verfügbarkeit eines IT-Systems gegenüber.

Dies gilt für alle Organisationen (Betreiber, für die Betriebssicherheit zuständige Behörden, Lieferanten) als auch für alle vom Bahnverkehr betroffenen Länder und allgemein auch für die von der Betriebssicherheit betroffene Industrie.

Die komplexen Industriesysteme verbinden sowohl physische Funktionen als auch informatische Funktionen. Es wäre demnach nützlich, eine allgemeine Methode zu definieren, die den Einfluss programmierter Geräte auf stochastische Sicherheitsanalysen abschätzt, wenn die zwei Konzepte kombiniert werden.

Um eine adäquate Antwort auf diese Frage zu geben, müssen erst noch Hürden überwunden werden insbesondere:

- La vérification formelle des équipements programmés, et en particulier de leurs logiciels. Les méthodes de vérification «classiques» sont essentiellement basées sur l'examen des processus de développement, sur des tests, et sur des revues de conception et de codage. Les techniques de vérification formelle, maintenant accessibles, permettent de détecter des défauts de programmation et de spécification. Certaines méthodes permettent d'identifier toutes **les séquences d'entrées conduisant à un état redouté donné**, permettant ainsi d'en chiffrer l'occurrence.

- L'amélioration de la qualité des spécifications fonctionnelles. Les experts en sûreté de fonctionnement des équipements programmés le reconnaissent, que le point le plus faible du processus de développement des équipements programmés au niveau des exigences les plus élevées est la spécification fonctionnelle. Il convient donc d'améliorer significativement cette phase clé du développement.
- La quantification des probabilités[7] de défaillance « systématiques » des architectures redondantes programmés. Le problème est encore plus difficile pour les probabilités de défaillance corrélée de plusieurs équipements.
- La prise en compte du vieillissement des circuits et cartes électroniques dans les modèles, du fait notamment des évolutions technologiques en électronique (intégration de plus en plus poussée avec des géométries de plus en plus fines).
- La représentation de l'impact du contrôle commande et des équipements programmés sur la disponibilité, et non plus seulement sur la fiabilité, d'une installation.

- Die formale Überprüfung der programmierten Geräte und insbesondere deren funktionelle Software. Die „klassischen" Überprüfungsverfahren beruhen im Wesentlichen auf der Prüfung der Verfahren der Entwicklung, auf Testfällen und auf Design- und Kodifizierungskonzepten. Die Techniken der formalen Überprüfung sind heutzutage verfügbar und sie erlauben die Erkennung von Programmier- und Spezifizierungsfehler. Bestimmte Verfahren erlauben es, **alle Eingangssequenzen zu identifizieren**, die zu einem bestimmten befürchteten Zustand führen und deren Auftreten zu quantifizieren.
- Die Verbesserung der Qualität der funktionellen Spezifikationen. Die Experten der Betriebssicherheit programmierter Geräte sehen ein, dass bei funktionellen Spezifikationen der schwächste Punkt des Entwicklungsverfahrens programmierter Geräte auf der Ebene der wichtigsten Anforderungen liegt. Diese Schlüsselphase der Entwicklung muss also bedeutend verbessert werden.
- Die Quantifizierung „der systematischen" Ausfallwahrscheinlichkeit[8] redundant programmierter Architekturen. Das Problem ist bei korrelierten Fehlerwahrscheinlichkeiten verschiedener Geräte noch schwieriger.
- Die Berücksichtigung der Alterung der elektronischen Schaltungen und Karten in den Modellen, insbesondere aufgrund der technologischen und elektronischen Entwicklungen (immer stärkere Integration).
- Die Darstellung des Einflusses der Steuerung und der programmierten Geräte auf die Verfügbarkeit - und nicht mehr nur auf die Zuverlässigkeit - einer Anlage.

[7] Aspect probabiliste défini à partir des probabilités d'occurrence de tout ou partie des entrées externes menant à une des situations dangereuses.

[8] Wahrscheinlichkeitsaspekt, der auf der Auftrittwahrscheinlichkeit von allen oder von einem Teil der zu gefährlichen Situationen führenden externen Eingänge, basiert.

Ces deux approches, probabiliste et déterministe, sont complémentaires et peuvent s'alimenter l'une l'autre dans le cadre d'une démarche qui pourrait être la suivante :

1. Étude de sûreté système générale qui, outre ses résultats habituels, doit :
   - identifier les fonctions critiques réalisées par des systèmes informatisés ;
   - définir les postulats de fonctionnement relatifs à ces fonctions ;
   - définir les propriétés de sécurité que doivent remplir ces fonctions (prédicats, obligations de preuve…).
2. Étude de la sécurité du système informatisé qui se doit de :
   - chiffrer selon les méthodes classiquement retenues les taux de pannes sûre et non sûre imputables aux matériels supportant la fonction (en fonction de l'architecture notamment) ;
   - **d'identifier la liste exhaustive des séquences ordonnées des entrées qui conduisent à un des états considérés « non sûrs ». S'il n'en existe aucun, le taux de défaillance non sûre du fait du logiciel peut être considéré comme nul** ;
   - valoriser les probabilités conditionnelles d'occurrence des différentes séquences afin d'estimer les taux de pannes sûres et non sûres liées aux sollicitations possibles des fonctions critiques.
3. Compléter l'étude de sûreté système générale en intégrant les taux d'occurrence des comportements sûrs et non sûrs ainsi estimés.

Cette démarche n'est envisageable qu'avec une validation formelle du logiciel. **Certaines méthodes sont en mesure de décrire exhaustivement l'ensemble des séquences qui conduisent à des situations redoutées.**

Nous verrons ultérieurement que ce n'est généralement pas le cas des méthodes formelles les plus connues et que ce sera le cas de la méthode formelle que nous proposerons.

Diese zwei Ansätze, der stochastische und der deterministische, ergänzen sich gegenseitig im Rahmen folgender Vorgehensweise:

1. Allgemeine Sicherheitsanalyse des Systems; diese muss außer den üblichen Ergebnissen, auch noch folgendes erbringen:
   - Identifizierung der von IT-Systemen realisierten kritischen Funktionen
   - Definition der Anforderungen an die Funktionsweise dieser Funktionen
   - Definition der Sicherheitseigenschaften, die diese Funktionen erfüllen müssen (Prädikate, Beweispflicht…).
2. Sicherheitsanalyse des IT-Systems, die folgendes erbringen muss:
   - Quantifizierung der Ausfallraten, gemäß der üblichen Verfahren (sichere und unsichere Ausfälle) der zugrundeliegenden Hardware (insbesondere im Bezug auf die Architektur)
   - **Identifizierung aller ordnungsgemäßen Eingangssequenzen, die zu einem der als „unsicher" geltenden Zustände führen. Falls es keine gibt, kann die Rate der unsicheren Ausfälle der Software als null angenommen werden.**
   - Quantifizierung der bedingten Wahrscheinlichkeiten der verschiedenen Sequenzen um die Ausfallraten (sichere - unsichere Ausfälle) zu schätzen, die von einer möglichen Beanspruchung kritischer Funktionen herrühren
3. Vervollständigung der allgemeinen Untersuchung der Betriebssicherheit des Systems durch Integration der auf diese Weise geschätzten Raten sicheren und unsicheren Verhaltens.

Es ist anzumerken, dass dieses Vorgehen nur mit einer formalen Validierung der Software denkbar ist. **Bestimmte Verfahren sind in der Lage, alle Sequenzen, die zu befürchteten Situationen führen, zu beschreiben**.

Wie später gezeigt wird ist dies bei den bekanntesten formalen Verfahren im Allgemeinen nicht der Fall und steht im Gegensatz zu dem in dieser Arbeit vorgeschlagenen Verfahren.

Dans ce cas aussi, notons l'importance qu'il y a définir exhaustivement les :

1. postulats de fonctionnement, tant physiques, techniques que règlementaires ;
2. propriétés de sécurité attendues de l'installation dans son contexte ;
3. conditions d'indépendance entre l'architecture (matériel et logiciel de base) et le logiciel applicatif (aussi indépendant que possible de l'architecture) ;

À partir soit des expressions de besoins au plus haut niveau (fonctions d'enclenchement et de signalisation), soit du cahier des charges réputé correct remis par la maîtrise d'ouvrage.

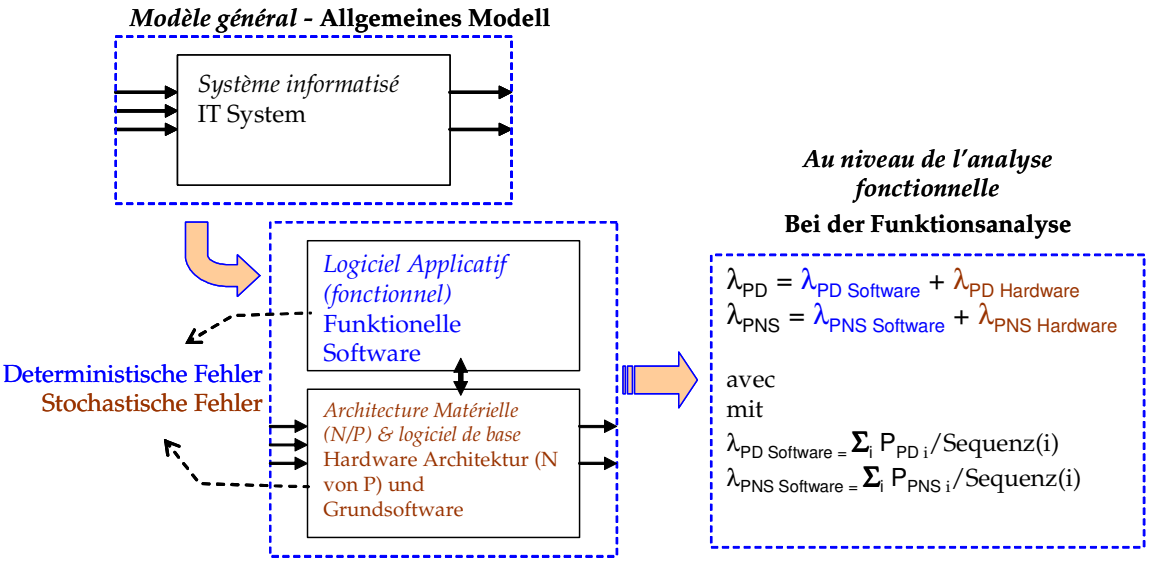
Figure 3.3 montre cette problématique.

Auch in diesem Fall ist die vollständige Definition folgender Punkte wichtig:

1. Anforderungen der technisch-physikalischen und vorschriftsmäßigen Funktionsweise
2. erwartete Sicherheitseigenschaften der Anlage und ihres Umfeldes
3. Unabhängigkeitsbedingungen zwischen der Architektur (Hardware und hardwareorientierte Software) und der Anwendungssoftware. Hier ist eine möglichst hohe Unabhängigkeit erwünscht.

Diese Definitionen erfolgen entweder in Funktion des Bedarfs auf höchster Ebene (Verschluss- und Signalisierungsfunktionen) oder auf Grundlage des als richtig geltenden Lastenheftes.

Abb. 3.3. veranschaulicht dies.




$\lambda_{PD}$ *(Taux d'occurrence d'une panne détectée ou panne sûre)* = $\lambda_{SSP}$ (Stündliche Auftrittsrate eines festgestellten oder sicheren Fehlers)

$\lambda_{PNS}$ *(Taux d'occurrence d'une panne non sûre)* = $\lambda_{SNSP}$ (Stündliche Auftrittsrate eines unsicheren Fehlers)

Les normes EN50126 et 50128 définissent les niveaux de sécurité à démontrer pour une application ferroviaire, dans notre cas pour l'architecture de base :

- SIL[9] 2 : Taux d'erreur non sûre entre $10^{-4}$ à $10^{-5}$/h - Niveau requis pour les systèmes de contrôle et commande
- SIL 4 : Taux d'erreur non sûre entre $10^{-8}$ à $10^{-9}$/h - Niveau requis pour les systèmes de sécurité

Dans les deux cas, le logiciel fonctionnel doit être spécifié et réalisé sans erreur.

Die EN5026 und EN50128 Normen definieren das Ziel Sicherheitsniveau für Eisenbahn Systeme. Dies kann für die Hardware und Grundsoftware gebraucht werden:

- SIL[10] 2: Die hinnehmbar unsichere Fehlerrate ist zwichen $10^{-4}$ und $10^{-5}$/ Stunde – Erforderlich für Betriebsfunktion
- SIL 4: Die hinnehmbar unsichere Fehlerrate ist zwichen $10^{-8}$ und $10^{-9}$/ Stunde - Erforderlich für Sicherheitsrelevante Signalfunktionen

In beiden Fällen muss die funktionelle Software ohne Fehler spezifiziert und entwickelt sein.

[9] SIL : Safety Integrity Level

[10] SIL : Safety Integrity Level

## 3.5 Maintenabilité et modifiabilité

La modifiabilité d'une application informatique est son aptitude à pouvoir être mise à jour, affinée et corrigée par les experts du métier, sans un recours prédominant aux experts informatiques.

La maintenabilité d'une application informatique est son aptitude à pouvoir être modifiée et testée rapidement et à moindres coûts économiques et temporels (délais de remise en conditions opérationnelles). La maintenabilité et la modifiabilité du fonctionnel de systèmes informatiques reposent avant tout sur :

- l'expression des fonctionnalités au moyen d'un langage métier suffisamment graphique et intuitif pour être abordé sereinement par les experts du métier. Il est à noter que l'expérience de la SNCF a montré que la modélisation des fonctionnalités métiers dans des langages analytiques n'a soit pas pu aboutir, soit n'a pas pu permettre un maintien dans le temps des modélisations réalisées [SNCF, 2005] [Bombardier, 2005] ;
- la distinction stricte d'une part, des aspects logiciels et matériels assurant les fonctions indépendantes de l'application proprement dites et, d'autre part, des aspects fonctionnels de l'application indépendamment des autres aspects que l'on peut qualifier d'informatiques. Dans ce contexte, une évolution des logiciels fonctionnels n'a aucun impact sur les logiciels gérant le matériel et la sécurité et inversement ;
- la démonstration de sécurité inhérente à toute modification de l'ensemble matériel et logiciel système doit être indépendante des technologies matérielles (type de processeur…). C'est un point clé pour la maintenance, il permet alors leur remplacement sans remise en cause des démonstrations de l'atteinte des objectifs (non régression) requises au sens des normes en vigueur [EN50128, 2001 [ENR0126, 2000] [EN50129, 2003] [L108/4, 2009] et une importante économie pour le maintien en conditions opérationnelles des systèmes.

Il est à noter que la part des accidents faisant suite à des modifications d'un système informatique ou à des interventions de maintenance sont respectivement de 20 et 12% [Gartner, 2007] [Gartner, 2008].

## 3.5 Instandhaltbarkeit und Migration

Die Migration einer IT-Anwendung ist die Möglichkeit der Aktualisierung, der Verfeinerung und der Korrektur dieser Anwendung durch Spezialisten des Fachs, ohne vorwiegenden Rückgriff auf IT-Experten.

Die Instandhaltbarkeit einer IT-Anwendung ist die Möglichkeit zur raschen Änderung und zum raschen Testen der Anwendung und zwar ohne allzu großen Aufwand (Dauer bis zur Wiederherstellung des Betriebes). Die Instandhaltbarkeit und die Migrationsmöglichkeit beruhen vor allem auf folgendem:

- Beschreibung der Funktionalitäten in einer Expertensprache, die genügend graphisch und intuitiv ist, dass die Spezialisten sie ohne Probleme verstehen können. Dazu ist folgendes anzumerken: die SNCF Erfahrung zeigt, dass das Modellieren der Funktionalitäten in Fachsprache durch eine analytische Sprache entweder nicht gelungen ist, oder dass es nicht gelungen ist, die Modellierungen dauerhaft zu konzipieren. [SNCF, 2005] [Bombardier, 2005].
- Strikte Trennung der Soft- und Hardwareaspekte einerseits (zur Sicherung der unabhängigen Funktionen der Anwendung) und - der funktionellen Aspekte der Anwendung - und der IT-Aspekte andererseits. In diesem Zusammenhang hat die Evolution der funktionellen Software keinen Einfluss auf die Basissoftware, die die Hardware und die Sicherheit steuert, und umgekehrt.
- Die mit jeglicher Änderung des Systems (Hard- und Software) einhergehende Sicherheitsbeweisführung muss unabhängig sein von der Hardwaretechnologie (Art des Prozessors, usw.). Dies ist wichtig für die Instandhaltung: dadurch kann die Hardware ersetzt werden, ohne die auf Grund der geltenden Normen [EN50128, 2001 [ENR0126, 2000] [EN50129, 2003] [L108/4, 2009] notwendigen Beweisführungen und Zulassungen in Frage zu stellen. Dies führt zu deutlichen Einsparungen bei der Erhaltung des betriebsfähigen Zustandes der Systeme.

Es ist anzumerken [Gartner, 2007] [Gartner, 2008], dass Unfälle nach IT-Systemänderungen oder Instandhaltung jeweils 20 und 12% aller Unfälle ausmachen.

## 3.5.1 Correction des erreurs

Supposons une organisation simple, qui n'est pas nécessairement celle effectivement mise en place dans la conception d'un logiciel, mais qui peut l'être. Un logiciel complexe est découpé en sous-ensembles, sous-programmes, modules ou objets selon le type de programmation adopté. Ce sont des unités fonctionnelles qui peuvent atteindre un degré de complexité suffisant pour mobiliser une équipe de programmeurs. Éventuellement, une équipe peut se voir confier la programmation de plusieurs modules ou objets. Il est fréquent que le développement de certains modules soit confié à un sous-traitant. Considérons le traitement d'une anomalie constaté à un stade avancé du déploiement du logiciel :

La première phase du traitement est d'en identifier les causes. Une cause fréquente d'anomalie est ce qu'on appelle un effet de bord, c'est-à-dire un résultat inattendu pris par le résultat d'une fonction lorsque l'un des paramètres d'entrée atteint ou dépasse une valeur limite. Par exemple, une valeur codée sur un octet : elle ne devrait normalement prendre que des valeurs comprises entre 0 et 255. Mais, parce que l'équipe chargée du développement d'un module a négligé l'hypothèse que la valeur produite puisse dépasser 255, elle a oublié d'inclure un test de sortie. Apparaît alors une séquence d'entrées conduisant à ce que le module passe aux autres les valeurs… 255, 0… c'est à dire une discontinuité, au lieu de…255, 256. La valeur 256 a été écrêtée par une limite physique de la machine lorsqu'on lui demande de stocker une valeur sur un octet. Notons que nous avons ici un phénomène apparemment chaotique—**une brutale divergence entre le prévu et le réel dans une machine ordinairement non chaotique.** Les modules qui utilisent ces résultats produits par le premier s'en accommodent plus ou moins bien. Par exemple, un module qui ne s'intéresse qu'à la parité du résultat, n'est pas perturbé par cette valeur aberrante. En revanche, celui qui intègre le résultat dans un autre calcul ou dans un test, risque fort de produire lui-même un résultat aberrant, ou d'entrer dans une boucle infinie…

## 3.5.1 Eliminierung der Fehler

Es wird als erstes eine einfache Entwicklungsorganisation betrachtet. Eine komplexe Software wird, je nach Programmierung, in Bausteine, Unterprogramme, Module oder Objekte unterteilt. Es handelt sich dabei um funktionelle Einheiten, die einen ausreichenden Komplexitätsgrad erreichen können, um ein Programmiererteam zu beschäftigen. Gegebenenfalls kann einem Team die Programmierung mehrerer Module oder Objekte anvertraut werden. Die Entwicklung einiger Module wird häufig außerhalb der Firma durchgeführt. Im Folgenden wird der Fall beschrieben, bei dem ein Softwarefehler in einem fortgeschrittenen Stadium der Softwareentwicklung festgestellt und behoben wird.

Die erste Phase der Behebung besteht in der Ursachenidentifizierung. Eine häufige Ursache eines Programmierfehlers ist der so genannte Seiteneffekt. Dies bedeutet, dass eine Funktion ein unerwartetes Ereignis liefert, einer der Ausgangsparameter einen Grenzwert erreicht oder gar überschreitet. Ein Beispiel ist ein auf einem Byte kodierter Wert, der normalerweise nur Werte zwischen 0 und 255 annehmen darf. Da jedoch das Entwicklungsteam die Bedingung, wonach der erkannte Wert nie 255 überschreiten darf, vernachlässigt hat, hat es auch den Ausgangstest vergessen. Falls eine Eingangssequenz erscheint, die dazu führt, dass das Modul die Werte überschreitet, erzeugt das Modul einen Überlauf vom Wert 255 zum Wert 0, anstatt zum Wert 256 und erreicht so eine Unstimmigkeit. Der Wert 256 wird von einem physischen Postulat der Maschine begrenzt, falls man einen Wert auf einem Byte speichert. Das Ergebnis ist ein offensichtlich chaotisches Phänomen - **eine brutale Divergenz zwischen dem vorgesehenen und dem wirklichen Wert, in einer normalen nicht chaotischen Maschine**. Die Module, die mit diesen erzeugten Ergebnissen arbeiten, kommen mehr oder weniger gut mit diesen Werten zurecht. Zum Beispiel wird ein Modul, das sich nur für die Parität des Ergebnisses interessiert, nicht durch diesen abweichenden Wert gestört. Andererseits, das Modul, das das Ergebnis in eine andere Berechnung oder in einen Test integriert, läuft Gefahr, selbst zu einem falschen Ergebnis zu kommen oder in eine unendliche Schleife hineinzulaufen …

La seconde phase de traitement de l'anomalie est de trouver une solution. La difficulté est plus grande qu'il n'y paraît. Non pas en raison des choix techniques à faire —modifier le type de la variable, ajouter un test en sortie, ajouter des tests en entrée, que faire du résultat du test— mais en raison du travail que cela représente.

- A quel niveau l'anomalie doit-elle être corrigé?
- Est-ce le module utilisateur qui doit systématiquement tester les variables qu'on lui passe ?
- Est-ce le module producteur qui doit les tester avant de les passer à d'autres ?
- Que doit-on considérer comme l'origine réelle, première, de l'anomalie ?
- Qui doit faire ce travail ?

Si l'origine de l'anomalie «interne» au logiciel se situe dans un module développé par une équipe qui n'existe plus, ou par un sous-traitant qui n'est plus disponible ou qui demande un prix excessif pour effectuer la modification, le choix le plus rationnel sera peut-être d'appliquer un correctif à un autre endroit, voire dans l'organisation.

Ce schéma de correction des anomalies informatiques, volontairement simplifié, peut se retrouver sous des formes plus complexes dans la conception d'installations de sécurité. A ceci près que même si la technique n'est pas toujours l'informatique, les mêmes questions se posent.

D'une manière générale, il faut bien se garder de sauter tout de suite sur des solutions qui apparaissent en général assez rapidement, quand on fait une telle analyse. On identifie un maillon fragile, une lacune, spontanément on a la solution qui arrive. Alors là, attention, aux fausses bonnes idées…la solution s'insère dans l'ensemble du système ? Combien faut-il d'étages de défense pour aboutir à un système raisonnablement sûr ? Comment les tester ?

**Un système sûr est avant tout un système simple, bien conçu et validable exhaustivement. Aussi apparaît-il nécessaire de créer les conditions d'une validation formelle des fonctionnalités des logiciels pour de telles anomalie soient éradiqués au plus tôt.**

Die zweite Phase der Behebung des Fehlers besteht darin eine Lösung zu finden. Die Schwierigkeit ist größer, als es erscheint; nicht im Bezug auf die zu treffende technische Wahl – Typenänderung der Variablen, Hinzufügen von Eingangstests oder Verwendung der Ausgangswerte jedoch hinsichtlich des Arbeitsaufwands, den dies bedeutet.

- Auf welchem Niveau muss der Fehler behoben werden?
- Ist es das Benutzermodul, das systematisch die Variablen testen soll, die man ihm übergibt?
- Ist es das ausführende Modul, das die Variablen testen sollte, bevor es sie weitergibt?
- Was ist der wirkliche Ursprung des Programmierfehlers?
- Wer muss diese Arbeit durchführen?

Wenn sich der Ursprung des Fehlers in einem Modul befindet, das von einem Team entwickelt wurde, das nicht mehr besteht, oder durch ein externes Unternehmen, das nicht mehr verfügbar ist, oder das einen übermäßigen Preis verlangt, um die Änderung durchzuführen, ist die vernünftigste Wahl vielleicht, eine andere Stelle zu verbessern oder sogar die Art der Entwicklung der Software einer Firma zu verändern.

Dieses vereinfachte Schema der Verbesserung von Programmierfehlern findet sich in einer komplexeren Form in der Konzeption von Sicherheitsanlagen wieder. Auch wenn die verwendete Technik nicht immer die Informatik ist, stellen sich dieselben Fragen.

Wenn man eine solche Analyse durchführt, muss man sich im Allgemeinen hüten, sofort Lösungen anzunehmen, die sich als erstes anbieten. Man identifiziert ein schwaches Glied, eine Lücke, und kommt spontan auf eine Lösung. Vorsicht vor „genialen" Ideen… Fügt sich die Lösung in die Gesamtheit des Systems ein? Wie viel Aufwand ist notwendig, um ein sicheres System zu führen? Wie kann man es testen?

**Ein sicheres System ist vor allem ein einfaches System, gut geplant und komplett prüfbar. Auch erscheint es notwendig, die Bedingungen für eine formale Überprüfung der Software zu schaffen, damit solche Anomalien schon im frühesten Stadium beseitigt werden können.**

## 3.5.2 Essais avant mise en service

L'expérience montre [SNCF, 2005] [Lötschberg, 2007] [Bombardier, 2005] [Hoch, 2008] que les essais de système informatiques critiques ne sont pas exhaustifs, ce quel que soit le temps qui peut leur être alloué. En effet, même si une majorité des essais peuvent être réalisés en plateforme (hors site), la combinatoire est telle qu'une infime partie peut réellement être testée. Il s'agit notamment des essais négatifs[11], des commandes non prévues ou lancées dans des fenêtres temporelles particulières qui ne peuvent être testées, des concomitances d'événements extérieurs… ce alors qu'en pratique c'est essentiellement sur ces domaines que les systèmes informatiques présentent le plus de faiblesses. [SNCF, 2005] [Lötschberg, 2007] [Bombardier, 2005]

Il est à noter que la part des accidents dus à des erreurs (ou des incomplétudes) de spécification sont d'environ 44% [Gartner, 2009].

L'augmentation des possibilités de l'informatique industrielle conduit à une difficulté de validation, fait apparaître de plus en plus pressement le besoin d'une vérification *exhaustive* et *automatique* des conditions de sécurité requises par l'environnement et les conditions d'exploitation. Les équipes chargées des essais avant mise en service rencontrent des difficultés pour la réalisation des essais du fait de :

- la difficulté croissante de respecter les exigences des clients en terme de coûts et délais ;
- le fort renouvellement des équipes ;
- la difficulté croissante pour maintenir les compétences requises ;
- la complexité croissante des études des systèmes nouveaux ;
- le recours à de nombreux outils et méthodes, à la cohérence non garantie…
- la perception erronée que les systèmes informatisés sont aisément modifiables ;
- l'oubli par les concepteurs que le système informatique s'intègre dans un système global avec des hommes, des procédures pour gérer les modes dégradés.

[11] Essais visant à vérifier les indépendances fonctionnelles et/ou techniques admises implicitement

## 3.5.2 Versuche vor der Inbetriebnahme

Die Erfahrung [SNCF, 2005] [Lötschberg, 2007] [Bombardier, 2005] [Hoch, 2008] zeigt, dass Tests kritischer Rechnersysteme nicht vollständig sind. Es ist in der Tat so, dass selbst wenn die Mehrheit der Stellwerksversuche auf einer Plattform (und nicht am Standort) durchgeführt werden können, die Anzahl der Kombinationen derart groß ist, dass nur ein winziger Teil getestet werden kann. Es handelt sich insbesondere um negative Tests[12], um nicht vorgesehene bzw. um in speziellen Zeitabschnitten ausgeführte Befehle, die nicht getestet werden können, genauso wie die Simultanität fremder Ereignisse. Dabei ist es in der Praxis so, dass die Rechnersysteme gerade in diesen Bereichen die meisten Schwachstellen aufweisen [SNCF, 2005] [Lötschberg, 2007] [Bombardier, 2005].

Es ist anzumerken [Gartner, 2009] dass der Anteil Unfälle aufgrund falscher (oder unvollständiger) Spezifikation ungefähr 44% beträgt.

Die wachsenden Möglichkeiten der industriellen Informatik führen zu Schwierigkeiten bei den Tests und zeigen immer mehr die Notwendigkeit einer *vollständigen* und *automatischen* Überprüfung der aufgrund der Betriebsbedingungen notwendigen Sicherheitsbedingungen auf. Die mit den Versuchen vor der Inbetriebnahme beauftragten Teams sind aus folgenden Gründen mit Schwierigkeiten konfrontiert:

- steigende Probleme, die Kundenanforderungen in Sachen Kosten und Fristen einzuhalten,
- starke Fluktuation der Teams,
- Probleme, die notwendigen Kompetenzen zu erhalten,
- steigende Komplexität der Untersuchung neuer Systeme,
- Rückgriff auf zahlreiche Hilfsmittel und Verfahren, deren Kohärenz nicht gesichert ist,
- das Vernachlässigen der Tatsache durch den Entwickler, dass das Informatiksystem sich in ein globales System integriert mit Personen und Verfahren, die die Ausnahmezustände regeln.

[12] Tests um die impliziten funktionelle und technische Unhabhängigkeiten zu überprüfen,

## 3.6 Vers une architecture cible des automates de sécurité

Pour l'ensemble de ces raisons, le travail présenté reposera sur une architecture matérielle et logicielle telle qu'illustrée par la figure 3.4.

Il s'agit en fait de concevoir un automate de sécurité permettant de dissocier aisément les aspects matériel et logiciel de base ($\lambda_{PD}$, $\lambda_{PNS}$ pour le hard) et de ceux logiciel applicatif ($\lambda_{PD}$ et $\lambda_{PNS}$ pour le soft) (Cf. Figure 3.3).

Il s'agit en fait de revenir à l'approche «*fail safe*»[13] existant en France depuis le temps des postes d'aiguillages mécaniques et électromécaniques, de reconduire les fonctionnalités qui ont fait leurs preuves par l'expérience passée ainsi que les règlementations associées, pour obtenir des systèmes adaptatifs, maintenables, sûrs et donc économiques.

La figure 3.4 présente l'architecture cible avec les logiciels fonctionnels et de base indépendants.

## 3.6 Ansatz einer Zielarchitektur der Sicherheitsautomaten

Die Arbeit beruht auf einer Hard- und Softwarearchitektur gemäß nachstehender Abb. 3.4.
Ziel ist es, einen Sicherheitssautomaten zu konzipieren, der auf einfache Weise die Hardware und die hardwareorientierte Software ($\lambda_{PD}$, $\lambda_{PNS}$ für die Hardware) von der anwendungsbezogenen Software ($\lambda_{PD}$ und $\lambda_{PNS}$ für die Software) unterscheidet (Abb. 3.3).
Es handelt sich auch darum, zu dem in Frankreich seit der Zeit der mechanischen und elektromechanischen Stellwerke bestehenden „*fail safe*“[14] Konzept zurückzukommen, die Funktionen zu bewahren, die sich in der Vergangenheit bewährt haben, sowie die zugehörigen Regelungen, um adaptive, instandhaltbare, sichere und deshalb wirtschaftliche Systeme zu erhalten.

Die Abb 3.4 illustriert die Zielarchitektur mit unabhängiger anwendungsfunktioneller und Grundsoftware.

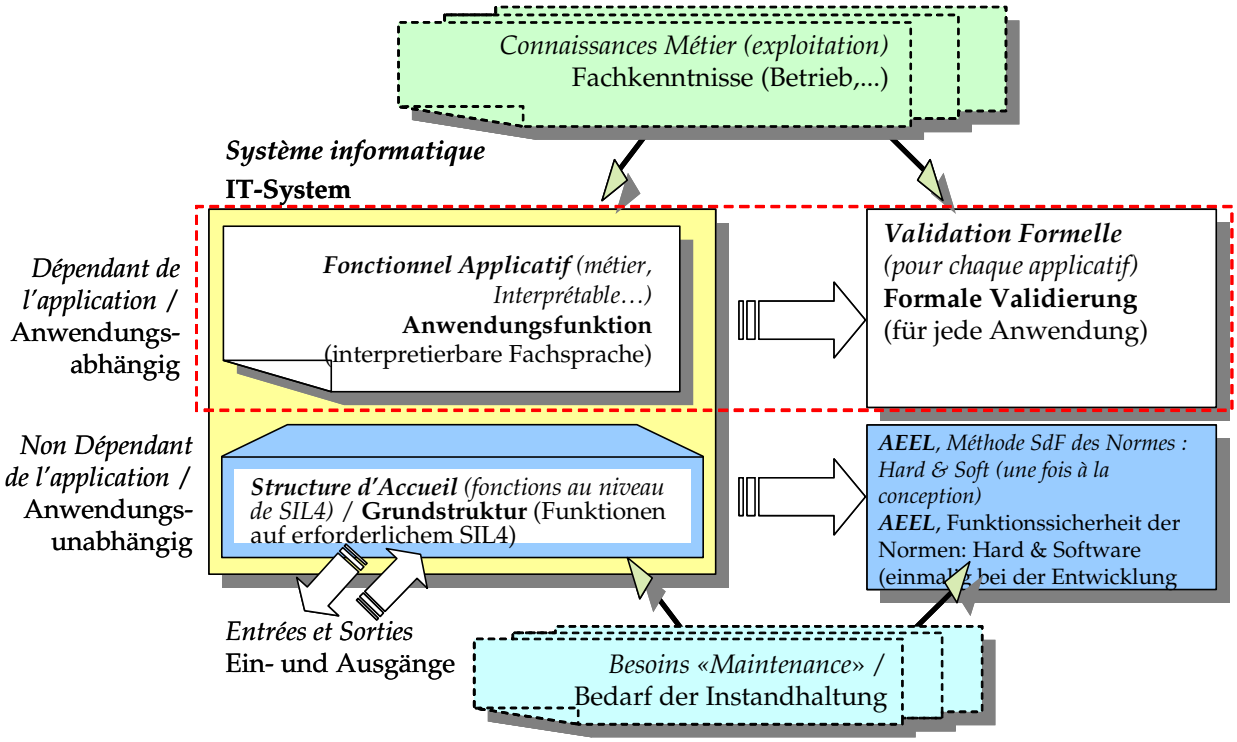



**Figure 3.4** : Vision théorique de l'architecture cible du système

**Abbildung 3.4**: Angestrebte Systemarchitektur

[13] Logique de sécurité où toute défaillance, perturbation extérieure ou sollicitation incorrecte entraîne une mise en position sûre (mode dégradé sûr).

[14] Sicherheitslogik, bei der jedes Versagen, jeder Störung oder fehlerhafte Benutzung eine Umstellung in eine sichere Position (sichere Rückfallebene) bewirkt.

**Commentaires:**

Dans la suite du travail nous allons utiliser pour les graphes décrivant le fonctionnel applicatif des notations particulières des réseaux de Petri.
Ces notations sont rendues possibles du fait de règles de conception des graphes :

- il existe un seul jeton par graphe ;
- il existe des indicateurs IND qui traduisent que le graphe occupe un sous ensemble des places du graphe ;

Ainsi si un fonctionnel applicatif recouvre deux graphes A et B et un indicateur Z, il vient que le vecteur d'état global s'écrit alors différemment de l'écriture habituelle, comme illustrée par la figure 3.5 :

**Bemerkungen:**

Im Zuge dieser Arbeit wird eine besondere kompakte PN-Schreibweise für die Beschreibung von Anwendungsfunktionen benützt.
Diese Beschreibung ist möglich weil:

- nur eine einzige Marke pro Graph existiert,
- Indikatoren existieren die zeigen ob einer von mehrem Plätzen eines Graphs besetzt sind oder nicht.

Wenn eine Anwendungsfunktion zwei Graphen A und B und einen Indikator benutzt, kann der Globalzustandsvektor wie in Abb.3.5 illustriert beschrieben werden.

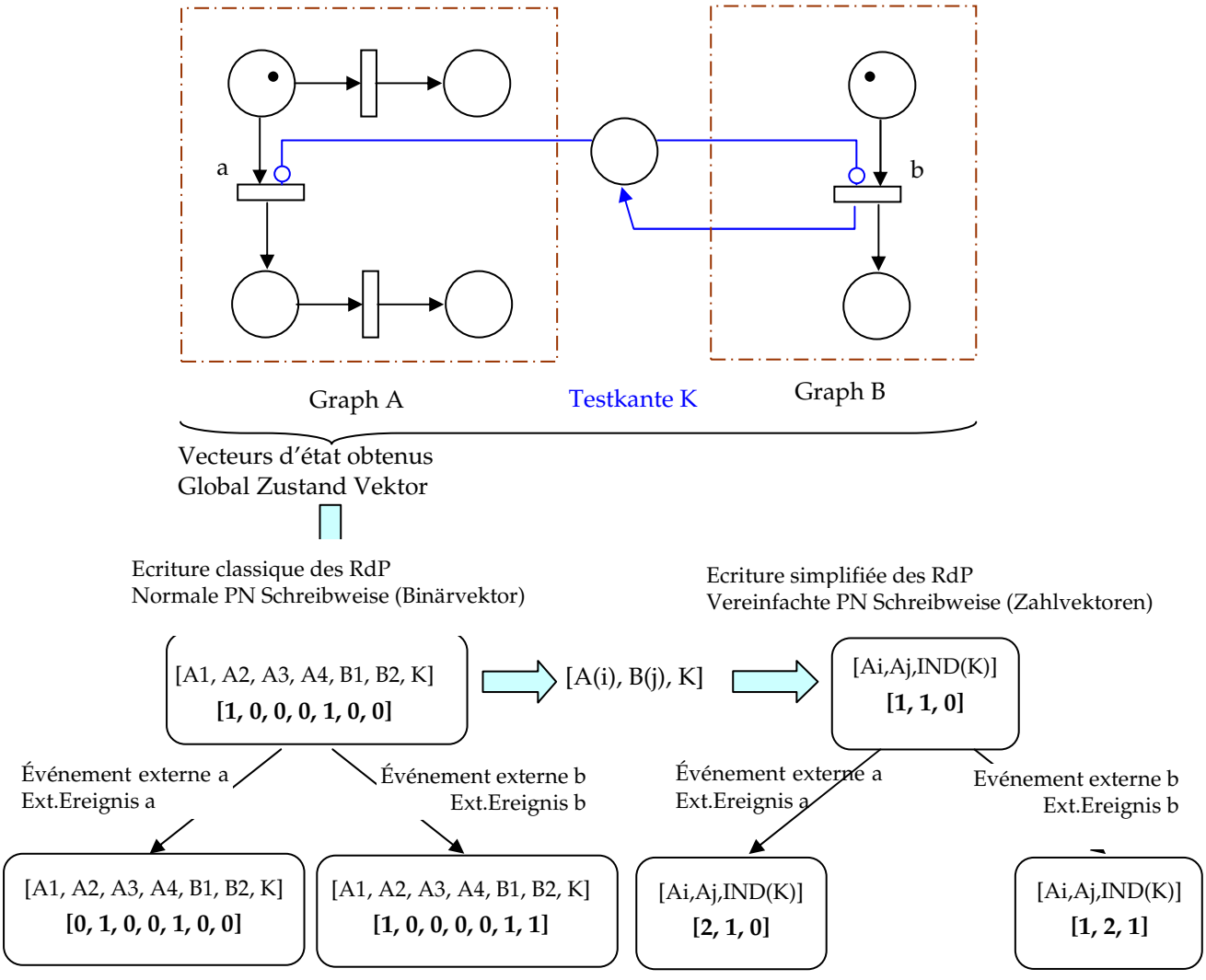



**Figure 3.5** : Détails de notations utilisées dans ce travail

**Abbildung 3.5**: Schreibweise in dieser Arbeit benützter Petrinetze

*NB: „Ereignis" wird hier als "ausserer" Zustand benützt.*

## 3.7 Ce qu'il faut retenir pour la suite du travail

Mémorisons les points clés suivants :

- les essais actuels ne peuvent conduire à un système informatique à haut niveau de sécurité compte tenu de la complexité croissante de cette tâche et des contraintes de coûts et de délais ;
- nécessité de dissocier dans les architectures des modules SIL4 les aspects „hard et logiciel de base" des aspects „logiciel fonctionnels" afin que ces modules soient validables ;
- la première couche „hard et logiciel de base" du module SIL4 est du ressort des approches probabilistes de par la prédominance matérielle et architecturale ;
- la seconde „logiciel fonctionnel" du module SIL4 est du ressort du déterminisme, des validations formelles (dans ce cas les architecture n'apportent aucune garantie) ;
- au niveau d'un système mixte, les deux approches sont complémentaires : ainsi l'approche probabiliste permet d'identifier les fonctions critiques réalisées en informatique (les conditions d'usage à risque, d'environnement…), l'approche déterministe fournit la liste exhaustive des « séquences » conduisant aux situations redoutées (permettant ainsi de créer un arbre de défaillance pour ces événements redoutés) afin d'en permettre la quantification de l'occurrence. Ces taux de défaillances peuvent alors être fournis par le modèle probabiliste initial ;
- en tout état de cause, vu la croissance rapide de la complexité des systèmes critiques modernes, de leur centralisation (satisfaisant les concepteurs en réduisant les interfaces, accablant les mainteneurs) les méthodes actuelles ne sont plus satisfaisantes, sauf à admettre une part d'aléatoire dans leur fonctionnement.

## 3.7 Zusammenfassung für die weitere Arbeit

Folgende Schlüsselaspekte sind festzuhalten:

- Aufgrund der steigenden Komplexität, der Kosten und der Verkürzung der Fristen können die heutigen Testverfahren nicht zu einem IT-System mit hohem Sicherheitslevel führen
- In der Architektur der SIL4-Module muss man unterscheiden zwischen „Hardware und hardwarebezogener Software" und „funktioneller Software", damit diese Module auch validiert werden können
- Aufgrund der Dominanz der Hardware und der Architektur unterliegt die erste Ebene („Hardware und hardwarebezogene Software") des SIL4-Moduls stochastischen Ansätzen
- Die zweite Ebene („funktionelle Software") des SIL4- Moduls unterliegt deterministischen Ansätzen und formalen Validierungen (die Architektur ist in diesem Fall keine Gewährleistung)
- Auf der Ebene eines gemischten Systems ergänzen sich beide Ansätze: mit dem stochastischen Ansatz werden kritische IT-Funktionen (risikoreiche Bedingungen der Verwendung, des Umfeldes, usw.), identifiziert; mit dem deterministischen Ansatz erhält man die komplette Liste der zu den befürchteten Ereignissen führenden „Sequenzen" (ermöglicht die Aufstellung des Fehlerbaums der befürchteten Ereignisse), um deren Häufigkeit festzustellen. Die Fehlerraten berechnet man mit dem ursprünglichen stochastischen Modell
- Auf jeden Fall sind die heutigen Verfahren aufgrund der rapide ansteigenden Komplexität der modernen, kritischen Systeme und ihrer Zentralisierung (zur Zufriedenheit der Entwickler aufgrund der Schnittstellen-reduzierung, jedoch zum Leid des Instand-haltungspersonals) nicht mehr zufrieden-stellend, es sei denn, man akzeptiert eine zum Teil zufallsbedingte Funktionsweise.

# CHAPITRE 4

# Identification des principes fondamentaux de la sécurité ferroviaire et état de l'art

L'application de méthodes formelles aux postes d'aiguillage requiert l'explicitation complète des propriétés de sécurité et des postulats de fonctionnement. Ce chapitre a pour objet de décrire le contexte ferroviaire afin d'en percevoir et formaliser les propriétés de sécurité, logiques et physiques, pour les systèmes informatiques critiques.

## 4.1 Sécurité ferroviaire - Notions fondamentales

La démonstration de la sécurité d'un système technique, informatique en l'occurrence, ne peut s'opérer sans tenir compte de son intégration dans son environnement, dans le système auquel il participe.

### 4.1.1 Sécurité et disponibilité

La notion de sécurité n'a pas les mêmes implications pratiques selon la nature de l'activité considérée. La sécurité des circulations ferroviaires repose notamment — et même principalement — sur la **possibilité de les arrêter**. Quand plus aucun train ne roule, tout danger lié à la circulation en elle même est écarté. D'où le système d'équations de base :

- **train arrêté = sécurité (indisponibilité) ;**
- **train en mouvement = danger (disponibilité).**

Il existe d'ailleurs une procédure d'urgence, le signal d'alerte radio qui commande à tous les trains de la zone où ce signal est émis de s'arrêter.

La littérature ne distingue presque jamais sécurité et fiabilité [EN50126, 2000] [L108/4, 2009] [2004L0049]. Cette relation étroite, pour ainsi dire causale, entre l'une et l'autre n'est presque jamais questionnée. **La fiabilité fait la sécurité ; le danger vient des défaillances [Schnieder, 2008]**.

# KAPITEL 4

# Sicherheitseigenschaften im Eisenbahnwesen und Stand der Technik

Die Anwendung formaler Methoden auf Eisenbahnstellwerke erfordert die vollständige Formalisierung der Sicherheitseigenschaften und der Funktionsanforderungen. Dieses Kapitel beschreibt das Eisenbahnwesen, um es zu ermöglichen, die logischen und physikalischen Sicherheitseigenschaften für kritische Computersysteme zu verstehen und zu formalisieren.

## 4.1 Bahnsicherheit - Grundbegriffe

Der Nachweis der Sicherheit eines technischen Systems, in dem vorliegenden Fall eines IT-Systems, muss die Einbindung in das betreffende Umfeld berücksichtigen.

### 4.1.1 Sicherheit und Verfügbarkeit

Je nach Art des betrachteten Bereichs hat der Sicherheitsbegriff nicht dieselben praktischen Auswirkungen. Die Sicherheit des Bahnbetriebs beruht insbesondere — und sogar im Wesentlichen — auf der Möglichkeit, die Fahrten auch zu unterbrechen. Wenn kein Zug mehr fährt, dann ist auch jede Gefahr in Verbindung mit den Zugbewegungen beseitigt. Daher rührt das System der Basisgleichungen:

- **stehender Zug = Sicherheit (Unverfügbarkeit),**
- **fahrender Zug = Gefahr (Verfügbarkeit).**

Es gibt im Übrigen eine Notvorschrift, die besagt, dass bei einem Funkwarnsignal alle Züge in dem Bereich, in dem das Signal abgegeben wird, anhalten müssen. In der Literatur wird fast nie zwischen Sicherheit und Zuverlässigkeit unterschieden [EN50126, 2000] [L108/4, 2009] [2004L0049]. Die enge, sozusagen kausale Verbindung zwischen den beiden Begriffen wird so gut wie nie in Frage gestellt. **Aus der Zuverlässigkeit ergibt sich die Sicherheit und die Gefahr kommt von den Ausfällen [Schnieder, 2008].**

Ce qui est discrètement évacué avec la notion de fiabilité, c'est que tous les évènements que l'on appelle défaillances ne sont pas également indésirables. Mais n'y a-t-il pas des défaillances plus souhaitables que d'autres ? Le problème est immédiatement transféré sur la sécurité : puisqu'on ne sait pas réaliser la fiabilité absolue.

Les ingénieurs sont toujours confrontés au manque de fiabilité des machines. Quelles qu'en soient les raisons, c'était un fait.

En 1949 que fut énoncée la fameuse loi de Murphy : **«s'il est possible que quelque chose se passe mal, alors cela arrivera» [Poncet, 2008]. Cette loi transforme une probabilité en certitude.**

Elle pose l'équation : la probabilité d'un évènement défavorable est égale à 1. Le délai d'occurrence importe peu : le nombre d'événements est suffisamment grand pour que la probabilité d'événement défavorable dans un délai limité, soit effectivement très élevée.

Il faut assurer la sécurité. Mais que faire quand les machines souffrent d'un manque de fiabilité ?

Pour ce faire il a été retenu en France de faire appel à quelques principes solidement établis, des lois physiques (pesanteur, Lenz…) que l'on peut considérer comme fiables, et sur lesquelles il est tentant d'appuyer une certaine fiabilité, locale, des composants. Par exemple, qu'un corps qui n'est soumis à aucune action mécanique autre que son poids, tombe.

**Ce qu'il faut retenir, c'est que la sécurité telle qu'elle a été conçue, ne repose pas alors sur la fiabilité du dispositif, mais sur la fiabilité de quelques lois physiques et sur une conception « orientée » des machines.**

Orientée, dans le sens où toutes les évolutions possibles du mécanisme ne sont pas équivalentes. Elles sont hiérarchisées et exploitées selon les principes suivants :

1. Certains états sont plus stables que d'autres ;

2. L'évolution «spontanée» du sous-système se fait toujours vers l'état le plus stable ;

3. Il est possible de faire correspondre l'état le plus stable à la situation la plus sécuritaire.

Le «système» associant les systèmes techniques ont été conçus de façon qu'aucun des résultats de l'épreuve de leur fonctionnement n'entraîne la réalisation d'un événement «contraire à la sécurité».

Beim Begriff der Zuverlässigkeit bleibt folgendes auf diskrete (digitale) Weise unbeachtet: nicht alle als „Ausfälle" bezeichneten Ereignisse sind auf die gleiche Weise unerwünscht. Das Problem wird sofort auf die Sicherheit übertragen, da es keine absolute Zuverlässigkeit gibt.

Die Ingenieure sind immer der mangelnden Zuverlässigkeit der Maschinen ausgesetzt, was auch immer die Gründe dafür sein mögen. Das ist eine Tatsache.

Das bekannte „Murphy-Gesetz" stammt aus dem Jahr 1949: **„was auch immer schief gehen kann, wird auch schief gehen" [Poncet, 2008]. Durch dieses Gesetz wird die Möglichkeit zur Gewissheit.**

Hieraus ergibt sich folgende Gleichung: die Wahrscheinlichkeit eines ungünstigen Ereignisses ist gleich 1, der Auftrittszeitpunkt spielt fast keine Rolle, die Anzahl der Ereignisse ist genügend groß, dass die Wahrscheinlichkeit des Eintretens eines ungünstigen Ereignisses innerhalb eines bestimmten Zeitraums sehr hoch ist.

Die Sicherheit muss gewährleistet werden, aber was kann man machen, wenn die Maschinen nicht zuverlässig genug sind?

In Frankreich stützt man sich in dem Fall auf physikalische Gesetze (Schwerkraft, elektromagnetische Induktion…); diese gelten als zuverlässig und es bietet sich an, eine bestimmte lokale Zuverlässigkeit der Komponenten darauf zu stützen. Ein Körper, der keiner anderen mechanischen Aktion außer seinem Gewicht ausgesetzt ist, fällt.

**Folgendes ist festzuhalten: die Sicherheit, wenn sie auf diese Weise konzipiert wird, beruht nicht auf der Zuverlässigkeit der Anlage, sondern auf der Zuverlässigkeit einiger physikalischer Gesetze und auf einer „orientierten" Gestaltung der Maschinen.**

Orientiert insofern, als dass alle möglichen Veränderungen der Zustände des Mechanismus nicht gleichwertig sind: es gibt eine Hierarchie und die Zustände werden gemäß nachstehenden Prinzipien genutzt:

1. Einige Zustände sind stabiler als andere.

2. Die „spontane" Veränderung des Teilsystems geht immer in Richtung des stabilsten Zustandes.

3. Es ist möglich, den stabilsten Zustand als den sichersten zu definieren.

Das alle technischen Systeme umfassende „System" ist derart konzipiert, dass keines der Ergebnisse des Funktionstests zu einem „sicherheitswidrigen" Ereignis führt.

Cette conception de la sécurité, distincte de la fiabilité, a cependant une contrepartie qui n'est pas gratuite : une maintenance préventive rigoureuse. Car il ne suffit pas qu'une installation se mette d'elle-même dans un état « sécuritaire » lorsqu'elle tombe en panne, pour que la sécurité soit assurée.

Encore faut-il que les opérateurs, sous la contrainte de la production, ne soient pas poussés à s'affranchir d'une installation trop souvent en défaut. **Ce modèle de la sécurité ne repose donc pas seulement sur la technique, malgré les apparences**[15]. **Ainsi, l'homme, dans le cadre des procédures, joue le rôle principal dans le système ferroviaire** [Reason, 1993] [Reason, 1995] [SNCF, 1995].

La sécurité dans les chemins de fer français repose historiquement sur le **déterminisme**. Tout effet a une cause et si un effet est indésirable, en supprimer les causes doit permettre de l'éviter ! Raisonner dans un cadre déterministe impose de trouver une explication à tout et en particulier à ce qui, de prime abord, semble inexplicable. Et surtout, il s'agit de faire en sorte que l'événement ne se reproduise pas.

**Ce but peut être très près du «risque zéro», même si chacun s'accorde à penser qu'il n'existe pas.** Après tout, le cercle parfait non plus n'existe pas. Il fut cependant, et reste une idée très féconde… Le système est conçu en respectant des règles pratiques de conception :

- une cause unique ne saurait conduire à un accident,
- les défaillances potentielles vont toujours dans le sens de la sécurité,
- limiter les actions (y compris effets de la résilience) des hommes, procédures et outils,
- une définition précise du rôle de chacun des acteurs de la sécurité et une formation adaptée,
- des boucles de récupération indépendantes autant que faire se peut entre les différents acteurs, les procédures et les outils,
- un contrôle, si possible permanent, des hommes, des procédures et des outils,
- une amélioration permanente basée sur les retours d'expérience.

Dieses von der Zuverlässigkeit getrennte Sicherheitskonzept verlangt dafür jedoch eine sorgfältige, vorbeugende Instandhaltung, die nicht kostenneutral ist. Denn es genügt nicht, dass eine gestörte Anlage von selbst in einen „sicheren" Zustand übergeht, um die Sicherheit zu gewährleisten.

Es muss auch gewährleistet sein, dass das unter Arbeitsdruck stehende Bedienungspersonal nicht dazu verleitet wird, auf die zu oft ausfallende Anlage zu verzichten. **Entgegen dem Anschein beruht also dieses Sicherheitsmodell nicht nur auf der Technik**[16]**. Im Rahmen der Vorschriften spielt der Mensch somit die Hauptrolle im Bahnsystem** [Reason, 1993] [Reason, 1995] [SNCF, 1995].

Beim französischen Bahnsystem beruht die Sicherheit von jeher auf **Determinismen**. Es gibt für jede Wirkung eine Ursache, und ist davon eine unerwünscht, so muss sie durch den Wegfall der Ursachen verhindert werden. Im Rahmen der deterministischen Überlegung muss es für alles eine Erklärung geben, insbesondere für Dinge, die anfangs unerklärlich scheinen. Es muss vor allen Dingen dafür gesorgt werden, dass das Ereignis nicht noch einmal auftritt. **Dieses Ziel kann sehr nahe beim „Nullrisiko" liegen, selbst wenn jeder weiß, dass es dies nicht gibt**… Das System wird unter Einhaltung praktischer Regeln entwickelt:

- Eine einzige Störung darf nicht zum Unfall führen.
- Potentielle Fehler gehen immer in Richtung Sicherheit.
- Aktionen (einschließlich der Widerstandsfähigkeit) der Menschen, Vorschriften und Werkzeuge müssen beschränkt sein.
- Die Rolle eines jeden Sicherheitsbeteiligten muss genau definiert sein, es muss eine angemessene Ausbildung geben.
- Nach Möglichkeit unabhängiger Notfallplan zwischen den verschiedenen Beteiligten, Vorschriften und Werkzeugen.
- Kontrolle des Menschen, der Vorschriften und der Werkzeuge; ständige Verbesserung mithilfe der Erfahrung.

[15] Le retour d'expérience sur les relais électromécaniques de sécurité confirme ce raisonnement : avoir un relais qui reste en position haute malgré une interruption de son alimentation, se produit en moyenne plusieurs milliards de fois moins souvent que le contraire. Et c'est l'homme, en l'occurrence un agent de maintenance, qui contraint de façon préventive ou corrective, l'automate que constitue une installation de signalisation à rester à l'intérieur de son domaine de fonctionnement.

[16] Die Erfahrungen mit den elektromechanischen Sicherheitsrelais bestätigen diesen Rückschluss: ein Relais, das trotz Stromunterbrechung in der oberen Stellung bleibt, kommt im Schnitt mehrere Milliarden Male seltener vor als das Gegenteil. Es ist der Mensch (in dem Fall der Wartungsbedienstete), der den Automaten (sprich die Signalanlage) vorbeugend oder korrektiv so einstellt, dass er innerhalb seines Funktionsbereichs bleibt.

La fiabilité générale de fonctionnement a une influence directe sur la sécurité de système. Plus le nombre de retards et d'incidents est limité, moins d'opérateurs et procédures seront demandés et les risques seront ainsi plus faibles.

Die allgemeine Funktionsintegrität beeinflusst die Systemsicherheit direkt. Je begrenzter die Anzahl von Verspätungen und Vorfällen ist, desto weniger werden das Bedienungspersonal und die Vorschriften beansprucht; somit sind auch die Risiken geringer.

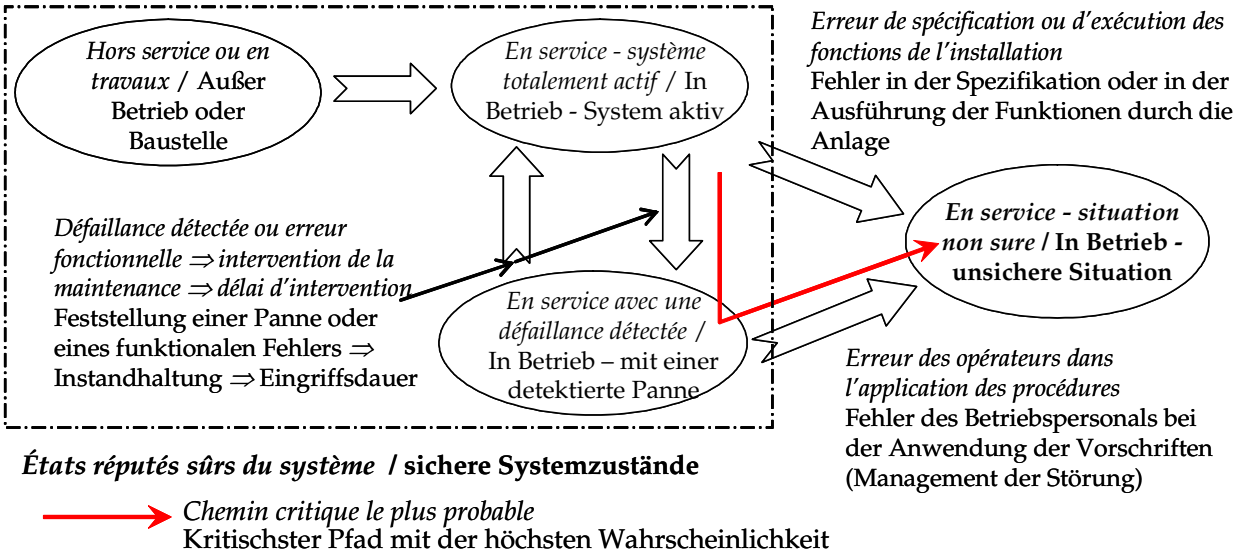



**Figure n°4.1**: Impact de l'indisponibilité sur le niveau de sécurité global du poste d'aiguillage informatique – en rouge le chemin le plus court (plus probable)

**Abbildung n°4.1**: Einfluss der Nichtverfügbarkeit auf das globale Sicherheitsniveau eines elektronischen Stellwerkes – in rot der kürzteste Weg (mit der höchsten Wahrscheinlichkeit)

Les installations de signalisation doivent aussi être sûres et rapidement réparables pour garantir le niveau de sécurité demandé : les deux chemins critiques doivent avoir des taux d'occurrence équivalents (cf. Figure n°4.1). La fiabilité de fonctionnement influe donc sur la sécurité du système. Plus le nombre d'incidents est réduit, moins les hommes, les outils et les procédures seront sollicités et les risques seront d'autant plus faibles [DI603, 1993].

Il est à noter [Reason, 1993] [SNCF, 1995] [Hollnagel, 2004] [Hollnagel, 2006] [Poncet, 2008] [Guarnieri, 2008] que le taux d'erreur dans l'application d'une procédure «non prévue» dans le cadre de la gestion d'un mode dégradé est sensiblement plus élevé que le même taux dans l'application d'une procédure dans un contexte normal d'exploitation.

Bahnanlagen müssen sicher sein und rasch repariert werden können, damit der erforderliche Sicherheitslevel gewährleistet werden kann: die zwei kritischen Wege (siehe Abb. 4.1) müssen die gleiche Häufigkeit haben. So beeinflusst die Systemzuverlässigkeit die Systemsicherheit. Je geringer die Anzahl der Vorfälle, desto weniger werden die Menschen, die Anwendungen und die Vorschriften beansprucht, so dass die Risiken niedriger sind [DI603, 1993].

Es ist anzumerken [Reason, 1993] [SNCF, 1995] [Hollnagel, 2004] [Hollnagel, 2006] [Poncet, 2008] [Guarnieri, 2008], dass die Fehlerquote bei der Anwendung eines „unvorhergesehenen Ereignisses" in der Rückfallebene deutlich größer ist, als die Fehlerrate bei der „unvorhergesehenen" Anwendung einer Vorschrift im normalen Betrieb.

L'histoire des facteurs humains dans les chemins de fer français est ancienne, même si l'on reste clairement dans le paradigme déterministe le plus radical. Ainsi les propos tenus en 1933 par Le Besnerais, ingénieur en chef de l'Exploitation de la Compagnie du Nord : «*Vous croyez avoir un bon aiguilleur : il est zélé, ponctuel, a de la mémoire : point capital. Mais il manque de sang froid : à un moment difficile il fera le mouvement qu'il ne fallait pas faire (…). Vous vous croyez innocent des victimes de la catastrophe : eh bien non, car vous avez mal choisi votre agent : il ne devait pas occuper ce poste : la psychotechnique vous l'aurait dit…* ». [Poncet, 2008]

Généralement, l'adjonction d'une installation de sécurité n'accroît le niveau de sécurité système que si les taux de défaillance « sûr » et « non sûr » sont suffisamment faibles pour compenser la réduction de la fiabilité humaine de l'opérateur dans l'application des procédures.

Ce constat est d'autant plus important qu'une telle adjonction n'est jugée utile que lorsque l'augmentation du trafic surcharge l'opérateur dont le taux d'erreur dans l'application des procédures est alors jugé inacceptable.

## 4.1.2 Sécurité intrinsèque et monde ferroviaire clos

Ce raisonnement déterministe n'est envisageable que dans un «**monde clos**». Le principe de clôture imprègne toutes les pensées en matière de sécurité industrielle.

**«La sécurité d'un système automatisé n'est envisageable que dans le cas d'un système clos; dès qu'il y a interaction, il y a risque d'imprévisible».** [Poncet, 2008]

Ce n'est pas tant l'interaction avec l'extérieur qui introduit le chaos, que la forme du système (installations techniques…), et subséquemment ce que cette interaction peut produire sur lui.

Ainsi, le principe d'une organisation isolée de son environnement constituait un fondement de la conception de la sécurité. La loi imposa même en 1846 le principe de clôture du domaine ferroviaire, y compris aux passages à niveau. Ce qui ne tarda pas à poser rapidement des problèmes organisationnels et financiers... Ainsi, pour le développement d'un système ferroviaire, les frontières doivent être définies clairement fonctionnellement, règlementairement et physiquement [Poncet, 2008].

Die Geschichte des menschlichen Faktors bei der französischen Eisenbahn ist lang, selbst wenn man sich ganz streng an die Definition hält. So zum Beispiel die Äußerungen von Le Besnerais, Chefingenieur beim Fahrbetrieb der Compagnie du Nord, im Jahre 1933: „*Sie gehen davon aus, dass Sie einen guten Weichensteller haben: er ist eifrig, pünktlich und hat ein gutes Gedächtnis; das sind wichtige Punkte, aber er ist nicht kaltblütig genug und in einer kritischen Situation wird er genau die Bewegung machen, die er nicht hätte machen dürfen (…). Sie meinen, im Hinblick auf die Opfer der Katastrophe unschuldig zu sein: eben nicht, da Sie Ihren Bediensteten schlecht ausgesucht haben: er hätte nicht auf dieser Stelle arbeiten dürfen: die psychotechnischen Analysen hätten Ihnen das gesagt…*“ [Poncet, 2008].

Im Allgemeinen erhöht eine zusätzliche Sicherheitsanlage das Sicherheitsniveau des Systems nur dann, wenn die „sicheren“ und die „unsicheren“ Ausfallraten niedrig genug sind, um die Reduzierung der menschlichen Zuverlässigkeit des Bedienungspersonals bei der Anwendung der Vorschriften auszugleichen.

Diese Feststellung ist umso wichtiger, als dass eine solche zusätzliche Anlage nur dann für nützlich gehalten wird, wenn der Verkehrsanstieg das Bedienungspersonal überfordert. Diese daraus resultierende Fehlerrate des Bedienungspersonals bei der Anwendung der Vorschriften wird dann als nicht akzeptierbar bewertet.

## 4.1.2 Inhärente Sicherheit und abgeschlossener Bahnbereich

Dieser deterministische Ansatz kann nur in einer „**geschlossenen Welt“** in Erwägung gezogen werden. Das Prinzip der „geschlossenen Welt“ prägt den ganzen Bereich der industriellen Sicherheit.

**„Die Sicherheit eines automatisierten Systems ist nur in einem abgeschlossenen System möglich; sobald es Wechselwirkungen gibt, gibt es auch Risiken von unvorhersehbaren Ereignissen.“** [Poncet, 2008]

Nicht so sehr die Wechselwirkung mit der Außenwelt führt zum Chaos, sondern die Systemstruktur (technische Anlagen, usw.) und, folglich, die Folgen der Wechselwirkungen.

Aus diesem Grund war das Prinzip der geschlossenen Organisation (vom Umfeld abgeschirmt) eine Grundlage bei der Entwicklung von Sicherheitskonzepten. 1846 wurde das Gesetz zum Einzäunen von Bahnbereichen erlassen, einschließlich der Bahnübergänge, was im Übrigen rasch finanzielle und organisatorische Probleme mit sich brachte [Poncet, 2008].

Trois types de propriétés doivent donc être impérativement et formellement définies :

- celles garantissant directement la «sécurité», du système, en propre et au travers de ses interactions avec son environnement ;
- celles garantissant la disponibilité opérationnelle, du système, en propre et par le biais de ses interactions ;
- celles décrivant les règles d'interaction avec son environnement.

Par ailleurs, un tel système doit, dans notre contexte, posséder au moins un état sûr. C'est un état dans lequel le risque issu d'un système est qualifié de sûr, position de repli sûre.

Il est convenu que le système se trouve dans un état sûr lorsqu'il est exploité dans les règles d'exploitation. L'état sûr est généralement un état de plus faible énergie en accord avec les règles d'exploitation. La position de repli sûre doit être accessible en peu de temps et en désactivant les fonctions du système.

### 4.1.3 Le traitement de la sécurité

Les caractéristiques techniques du chemin de fer permettent d'identifier et d'anticiper les conflits potentiels et imposent de mettre la sécurité au centre de toutes les actions de tous les acteurs. Globalement, la synthèse des unes et des autres a permis de construire un système très sûr. Compte tenu de leurs caractéristiques techniques et des leviers d'action disponibles, on distingue traditionnellement :

- la sécurité des circulations qui concerne les interactions des circulations entre elles ou le traitement des modifications inopinées de l'environnement interférant avec les circulations. Elle repose sur les opérateurs (Fig. 4.3) qui utilisent leurs outils et appliquent des procédures,
- la sécurité technique qui vise à permettre par la conservation des caractéristiques dimensionnelles et fonctionnelles à un train de circuler sur une infrastructure. Elle relève des opérateurs d'entretien des installations fixes et des opérateurs d'entretien du matériel roulant dont les interventions vis à vis des agents circulation et des conducteurs se font dans le cadre de procédures.

Für die Entwicklung eines Bahnsystems müssen die Grenzen funktionell, vorschriftsmäßig und physisch klar definiert werden.

Es müssen also dreierlei Eigenschaften unbedingt formal bestimmt werden:

- diejenigen, die die „Systemsicherheit" direkt und über Wechselwirkungen mit dem Umfeld gewährleisten,
- diejenigen, die die betriebliche Verfügbarkeit des Systems direkt und über Wechselwirkungen gewährleisten,
- diejenigen, die die Regeln der Wechselwirkungen mit dem Umfeld beschreiben.

Der in dieser Arbeit beschriebene Ansatz verlangt zudem, dass ein solches System mindestens über einen sicheren Zustand verfügt, bei dem das systembedingte Risiko als „sicher" gilt (sichere Rückfallebene).

Es wird festgelegt, dass das System sich dann in einem sicheren Zustand befindet, wenn es gemäß den Betriebsregeln betrieben wird. Der sichere Zustand ist im Allgemeinen ein mit den Betriebsregeln kompatibler Zustand geringerer Energie. Die sichere Rückfallebene muss kurzfristig zugänglich sein und die Systemfunktionen deaktivieren.

### 4.1.3 Sicherheitsmanagement

Die technischen Eigenschaften der Eisenbahn erlauben es, potentielle Konflikte zu identifizieren und ihnen zuvorzukommen; dazu ist es notwendig, dass die Sicherheit den Kern aller Aktionen aller Beteiligten bildet. Insgesamt führt die Synthese hiervon zu einem sehr sicheren System. Im Bezug auf die technischen Eigenschaften und die verfügbaren Handlungsmöglichkeiten wird üblicherweise folgendes unterschieden:

- die Fahrsicherheit: sie betrifft die Wechselwirkungen der Züge untereinander oder die Behandlung von unerwarteten Änderungen des Umfeldes, die den Zugverkehr beeinflussen. Sie stützt sich auf das Bedienungspersonal (Abb. 4.3) die die Anlage benützen und Vorschriften befolgen.
- die technische Sicherheit: durch das Aufrechterhalten der „dimensionalen" und funktionellen Eigenschaften erlaubt sie die Fahrt des Zuges auf der Infrastruktur. Sie stützt sich auf das Wartungspersonal der Bahnanlagen und der Fahrzeuge, wobei die Eingriffe, die den Fahrdienstleiter oder den Fahrzeugführer betreffen, Gegenstand von Vorschriften sind.

Toute modification d'un composant du système est donc susceptible de se répercuter sur un autre composant, et ainsi de mettre en cause l'équilibre de la sécurité du système lui-même. Il est donc nécessaire d'y veiller à toutes les phases de vie du système.

Les chemins de fer français ont dès l'origine pratiqué une **approche de type déterministe** [Pichon, 1886] [Descubes[17], 1898], dite de sécurité intrinsèque, qui suppose une connaissance exhaustive des modes de défaillance des constituants, possible tant que :

- les modes de défaillance sont déterminés par les caractéristiques physiques (gravitation universelle, mode de défaillance des composants…) ;
- la combinaison des défaillances reste accessible à l'analyse humaine ;
- la (première) panne est détectable par la mise en défaut du système.

Avec le développement des systèmes informatiques, systèmes sujets à pannes multiples ne conduisant pas systématiquement à la mise en défaut du système, les approches probabilistes et assurance qualité ont été préférées. Il est en effet impossible d'identifier et traiter toutes les causes possibles de défaillance non sûre.
L'approche probabiliste consiste à quantifier la probabilité d'occurrence et à évaluer les conséquences potentielles d'un événement redouté. Cette approche reste à manipuler avec précaution, car:

- le calcul comporte, généralement, d'importantes simplifications - les taux de défaillance prévisionnelles des composants sont mal connues et le nombre d'événements provenant du retour d'expérience est très faible;
- l'indépendance des événements, souvent admise pour faciliter le calcul, est rarement vérifiée dans la réalité comme le montrent de nombreuses catastrophes [Lötschberg, 2007] [SNCF, 2005] [Bombardier, 2005] [Hoch, 2008] [Poncet, 2008].

Il est à noter qu'une approche probabiliste ne peut s'appliquer sur les erreurs des logiciels, notamment découlant des erreurs de formalisation du fonctionnel applicatif et de ses traductions. Ainsi pour les systèmes informatiques critiques, ni l'approche probabilistes, ni l'approche assurance qualité ne sont acceptables.

Jede Änderung einer Systemkomponente kann also Rückwirkungen auf eine andere Komponente haben und somit das Gleichgewicht der Systemsicherheit in Frage stellen. Es ist daher notwendig, in jeder Lebensphase des Systems darauf zu achten.

Die französische Eisenbahn ist immer **deterministisch vorgegangen** [Pichon, 1886] [Descubes[18], 1898]: sie verfolgt die sogenannte inhärente Sicherheit, die die vollständige Kenntnis der Ausfallmodi der Komponenten erforderlich macht. Dies ist möglich, so lange:

- die Ausfallmodi von den physikalischen Eigenschaften (Schwerkraft, Ausfallmodi der Einzelkomponenten, usw.) bestimmt werden,
- die Kombination der Ausfälle noch vom Menschen analysiert werden kann,
- der (erste) Ausfall durch einen Systemfehler festgestellt werden kann.

Mit der Entwicklung von IT-Systemen, die vielfältigen Defekten ausgesetzt sind, die nicht unbedingt zu einem Systemfehler führen, hat man die stochastischen Ansätze und die Qualitätssicherung bevorzugt. Es ist in der Tat unmöglich, alle möglichen Ursachen von unsicheren Ausfällen zu identifizieren und zu behandeln.

Der stochastische Ansatz besteht in der Quantifizierung der Auftrittswahrscheinlichkeit und in der Bewertung der potentiellen Folgen eines befürchteten Ereignisses. Dieser Ansatz ist mit Vorsicht zu genießen, denn:

- im Allgemeinen beruhen die Berechnungen auf zahlreichen Vereinfachungen – die Ausfallraten der Komponenten sind nicht genau bekannt und die Anzahl bekannter Ausfälle ist sehr gering,
- die oft zur Vereinfachung der Berechnungen angenommene Unabhängigkeit der Ereignisse bestätigt sich selten in der Wirklichkeit, so wie es zahlreiche Katastrophen belegen. [Lötschberg, 2007] [SNCF, 2005] [Bombardier, 2005] [Hoch, 2008] [Poncet, 2008]

Ferner ist anzumerken, dass der stochastische Ansatz bei Softwarefehlern (und insbesondere bei solchen, die von Formalisierungsfehlern der Anwendungsfunktionen und deren Übersetzung herrühren) nicht gültig ist. Für kritische IT-Systeme ist weder der stochastische Ansatz noch der Ansatz der Qualitätssicherung akzeptabel.

[17] Descubles y dit notamment que l'application de sa méthode „offre une sécurité absolue que l'on obtient jamais complètement avec les méthodes ordinaires"

[18] Descubes sagt dass mithilfe seine Methode « kann man eine Absolutesicherheit erreichen, die man sonst nie bekommt »

C'est pourquoi il est nécessaire pour les gestionnaires d'infrastructure de trouver une approche méthodologique permettant de s'assurer exhaustivement de la correction du fonctionnel applicatif porté par les systèmes informatiques et ainsi de revenir au cas des défaillances déterministes (fail safe).

### 4.1.4 Démontrabilité de la sécurité d'un équipement informatisé

Ainsi, il apparaît les enseignements suivants pour le développement d'un tel système :

- le comportement de l'automatisme doit être déterministe, en fonctionnement normal tant qu'en mode dégradé ;
- les trois natures de propriétés doivent être formalisées sans ambiguïté :
  - prédicats de sécurité ;
  - prédicats fonctionnels (sur abondant) ;
  - postulats d'interaction avec l'environnement.
- le comportement de l'automatisme est dans ces conditions vérifiable exhaustivement.

Ceci peut se schématise par la Figure 4.2.

Deswegen ist es notwendig, dass die Infrastrukturbetreiber einen methodischen Ansatz finden, um sich von der vollständigen Richtigkeit der Anwendungsfunktionen der IT-Systeme zu vergewissern und so auf die Verfahren der deterministischen Ausfälle zurückzugreifen (fail safe).

### 4.1.4 Nachweisbarkeit der Sicherheit einer rechnerbasierten Anlage

Somit erscheinen folgende Punkte bei der Entwicklung eines rechnerbasierten Systems wichtig:

- das Verhalten des Automaten muss deterministisch sein, in normaler Funktionsweise wie auch in der Rückfallebene
- es müssen drei Eigenschaften eindeutig formalisiert werden:
  - Sicherheitsprädikate
  - funktionelle Prädikate (auch die überflüssigen Anforderungen)
  - Hypothese der Wechselwirkung mit der Umwelt
- das Verhalten des Automaten ist vollständig unter diesen Bedingungen nachprüfbar.

Dies kann wie in Abb. 4.2 schematisiert werden.

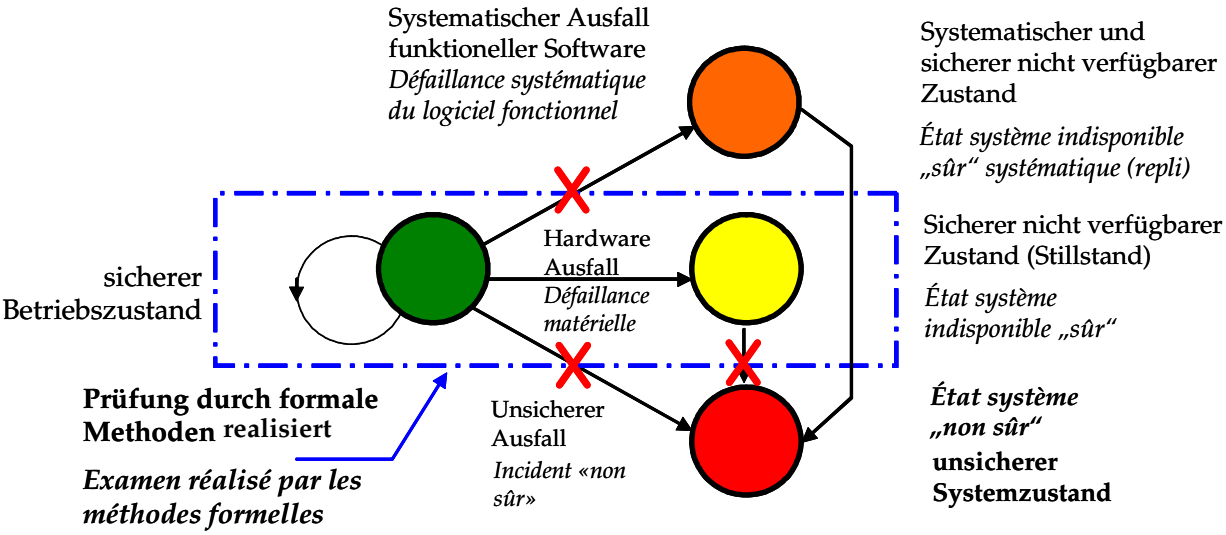



**Figure 4.2** : Etats systèmes traduisant les états fonctionnels et dysfonctionnels

**Abbildung 4.2**: Zustandsdiagramm eines inhärent-fail-safe-Systems

Les croix rouges traduisent les transitions dont il est nécessaire de prouver l'inexistence.

Dans ce cas, le système informatique fonctionnera de façon analogue à un système « fail safe ».

Die roten Kreise zeigen die verbotenen und Zustände, Schaltungen, welche die Überprüfungsmethode prüfen soll.

In diesem Fall wird das IT-System auf analoge Art wie in einem „fail safe" System funktionieren.

## 4.2 Généralités

Dans le mode routier, même si le code de la route lui impose de maîtriser son véhicule en toute circonstance, le conducteur est libre de sa direction.

Le chemin de fer est un mode de transport guidé où des véhicules équipés de roues en acier roulent sur des rails en acier. Le monde ferroviaire est un système à une seule dimension :

- où l'adhérence et le frottement sont faibles ce qui permet de faire circuler avec une faible dépense d'énergie des convois soit très lourds, soit très rapides ;
- où les changements de direction ne sont possibles que grâce à des installations spécifiques situées en des lieux définis.

Ces deux caractéristiques impliquent pour l'agent de conduite (ou mécanicien) :

- des distances de freinage très largement supérieures à celles permises par la vision directe de l'infrastructure[19] ;
- l'impossibilité de modifier sa route pour éviter une collision.

L'agent de conduite ne peut donc pas, en exploitation normale, conduire son train «à vue» et il n'a pas la maîtrise de la direction de son convoi. Il est donc nécessaire de mettre en place un exploitant sédentaire chargé de commander les installations fixes et d'organiser la circulation. Il s'agit pour l'agent d'exploitation des postes d'aiguillage de répondre à :

- la nécessité d'ordonnancer les circulations car les trains ne peut que se suivre entre deux installations successives où il sera possible de modifier leur succession ;
- la nécessité, généralement à l'aide d'installations techniques, de prévenir l'agent de conduite suffisamment tôt d'un ordre d'arrêt ou de ralentissement et de maintenir devant lui un espace libre d'une longueur suffisante pour lui permettre d'exécuter l'ordre transmis.

La règle de «maîtrise du véhicule par son conducteur» et le principe de «liberté de choix de la direction» ne s'appliquent donc pas au chemin de fer.

## 4.2 Bahnbetrieb

Im Straßenverkehr kann ein Verkehrsteilnehmer jederzeit frei seine Richtung wählen, auch wenn die Straßenordnung vorschreibt, dass er sein Fahrzeug jederzeit beherrschen muss.

Die Eisenbahn ist ein spurgeführtes Verkehrssystem, bei dem die Fahrzeuge mit Stahlrädern auf Stahlschienen rollen. Die Eisenbahn ist ein eindimensionales System, bei dem:

- Haftreibung und Reibung niedrig sind, wodurch sehr schwere oder sehr schnelle Züge mit wenig Energie verkehren können.
- ein Richtungswechsel nur dank spezifischer Anlagen und nur an ganz bestimmten Stellen möglich ist.

Beide Eigenschaften bedeuten für den Lokführer folgendes:

- Die Bremswege sind deutlich länger als diejenigen, die aus der Sicht der Infrastruktur[20] erlaubt sind.
- Er kann zur Vermeidung eines Zusammenstoßes unmöglich seinen Weg ändern.

Im normalen Betrieb kann der Eisenbahnfahrzeugführer also nicht „auf Sicht" fahren, er beherrscht die Fahrtrichtung des Zuges nicht. Deswegen ist es notwendig, Personal vorzusehen, das die festen Anlagen bedient und den Fahrbetrieb organisiert. Der Fahrdienstleiter muss:

- den Fahrbetrieb ordnen, denn die Züge können zwischen zwei Anlagen nur hintereinander fahren und erst mithilfe der Anlage ihre Reihenfolge ändern.
- den Eisenbahnfahrzeugführer früh genug vor einem Haltebefehl oder einem Langsambefehl vorzuwarnen - im Allgemeinen mittels technischer Anlagen, damit vor dem Zug ein genügend langer Abstand vorhanden ist, der die Ausführung des übermittelten Befehls ermöglicht.

Die Regel wonach „der Fahrer sein Fahrzeug beherrschen muss" und das Prinzip, wonach „er die freie Wahl seiner Richtung hat", gelten also bei der Eisenbahn nicht.

[19] Par rapport aux véhicules routiers, les distances d'arrêt possibles sont (elles ne sont pas utilisées en exploitation normale pour limiter les efforts sur les organes de frein) respectivement de l'ordre de 1000 m pour les trains de fret circulant à 100 km/h et de 3500 m pour les trains de voyageurs circulant à 300 km/h.

[20] Im Vergleich zu Straßenfahrzeugen beträgt der Bremsweg (dieser wird im normalen Betrieb nicht zur Begrenzung der Kräfte der Bremsorgane benutzt) um die 1000 m für mit 100 km/h verkehrende Güterzüge bzw. 3500 m für mit 300 km/h verkehrende Reisezüge.

L'exploitation ferroviaire consiste entre autre, sur une infrastructure, à assurer la gestion opérationnelle des circulations. L'exploitation apparaît comme l'articulation entre les moyens techniques existants et la réponse à donner aux demandes de la clientèle. En pratique, les composantes principales du système ferroviaire qui concourent à l'exploitation sont :

→ l'infrastructure :
- les rayons de courbure, les efforts admissibles sur les ouvrages d'art et les largeurs de plateforme déterminent les vitesses ;
- le nombre de voies, les pentes, le découpage des cantons influent sur le débit ;
- les rampes, la voie et les ouvrages d'art limitent les masses remorquables et donc nécessitent à volume de trafic égal plus de puissance de traction ou plus de trains…

→ le matériel roulant :
- la masse remorquable, la puissance, la vitesse maximale, la capacité de freinage influent fortement sur le débit ;
- le dimensionnement et le fonctionnement des portes facilitent les échanges des voyageurs et limitent les temps d'arrêt…
- La signalisation :
  C'est l'ensemble des moyens et installations permettant de tracer pour les trains des itinéraires «sûrs» et de transmettre les ordres aux conducteurs. La signalisation permet de parer à l'ensemble des risques résultant de la circulation des trains et de leur croisement avec les routes. On y regroupe les installations ayant trait à la sécurité technique des circulations à l'exclusion de ce qui concerne la voie, la plateforme et les ouvrages d'art.

**La conception et le maintien en condition opérationnelle des installations de signalisation n'autorise aucun trou d'enclenchement ou incertitude quant à la sécurité, qu'il s'agisse ou non de logiciels. Pour ces derniers il est donc impératif de trouver une voie donnant la garantie absolue que la réalisation est correcte.**

Beim Bahnbetrieb geht es unter anderem darum, dass auf einer bestimmten Infrastruktur der Fahrbetrieb gemanagt wird. Der Betrieb ist die Verbindung zwischen den bestehenden technischen Mitteln und der Kundennachfrage.

In der Praxis bestimmen folgende Eigenschaften des Bahnsystems den Bahnbetrieb:

→ Infrastruktur :
- Der Gleisradius, die zulässigen Kräfte auf Brücken und die Bahnkörperbreite bestimmendier Höchstgeschwindigkeit.
- Die Anzahl der Gleise, das Gefälle und der Blockabstand bestimmen die Verkehrsleistung.
- Die Steigung, die Gleise, die Brücken und die Tunnel begrenzen das Wagenzuggewicht und erfordern - bei gleichem Verkehrsvolumen - mehr Antriebsleistung oder mehr Züge.

→ Fahrzeuge:
- Das Wagenzuggewicht, die Leistung, die Höchstgeschwindigkeit und die Bremskapazität beeinflussen sehr stark die Verkehrsleistung.
- Die Breite und die Funktionsweise der Türen erleichtern das Ein- und Aussteigen der Fahrgäste und begrenzen so die Aufenthaltszeiten.
- Signalsystem:
  Es umfasst alle Mittel und Anlagen zur Stellung von „sicheren" Fahrwegen und zur Übermittlung von Befehlen an die Fahrzeugführer. Mit dem Signalsystem werden auch die an den Kreuzungen entstehenden Risiken begrenzt. Das Signalsystem umfasst alle Anlagen, die die technische Sicherheit des Fahrbetriebs betreffen mit Ausnahme der Gleise, des Bahnkörpers, der Brücken und der Tunnel.

**Die Entwicklung und die Funktionssicherheit der Signalanlagen erlauben keine „Sicherungslücken" und keine Ungewissheit im Hinblick auf die Sicherheit, ob es sich dabei nun um Software handelt oder nicht. Für die Software ist es also unbedingt nötig einen Weg zu finden, der die korrekte Ausführung uneingeschränkt gewährleistet.**

## 4.3 Le système ferroviaire

Comme tout processus industriel, un système comporte trois composantes: l'homme, les procédures et les installations [Bouvarel, 2003].

On ne peut définir de fonction de sécurité sans définir le contexte d'usage et d'environnement. Le système ferroviaire comprend des opérateurs au sol et à bord qui agissent sur une infrastructure et des matériels roulants en application d'une procédure dont les parties communes permettent de définir les interactions entre eux.

On peut le schématiser par la figure 4.3.

## 4.3 Bahnsystem

Wie jedes industrielle System besteht auch ein Eisenbahnsystem aus drei Teilen: Menschen, Vorschriften und Anlagen [Bouvarel, 2003].

Man kann keine Sicherheitsfunktion definieren, ohne dass man die Art der Verwendung und das Umfeld definiert. Das Bahnsystem umfasst Betriebspersonal vor Ort und an Bord der Züge; diese steuern mit Hilfe einer Infrastruktur die Fahrzeuge, sie wenden Vorschriften an, deren für alle gemeinsam geltenden Teile es erlauben, die Wechselwirkungen untereinander zu definieren.

Dies kann man wie in Abbildung 4.3 darstellen.

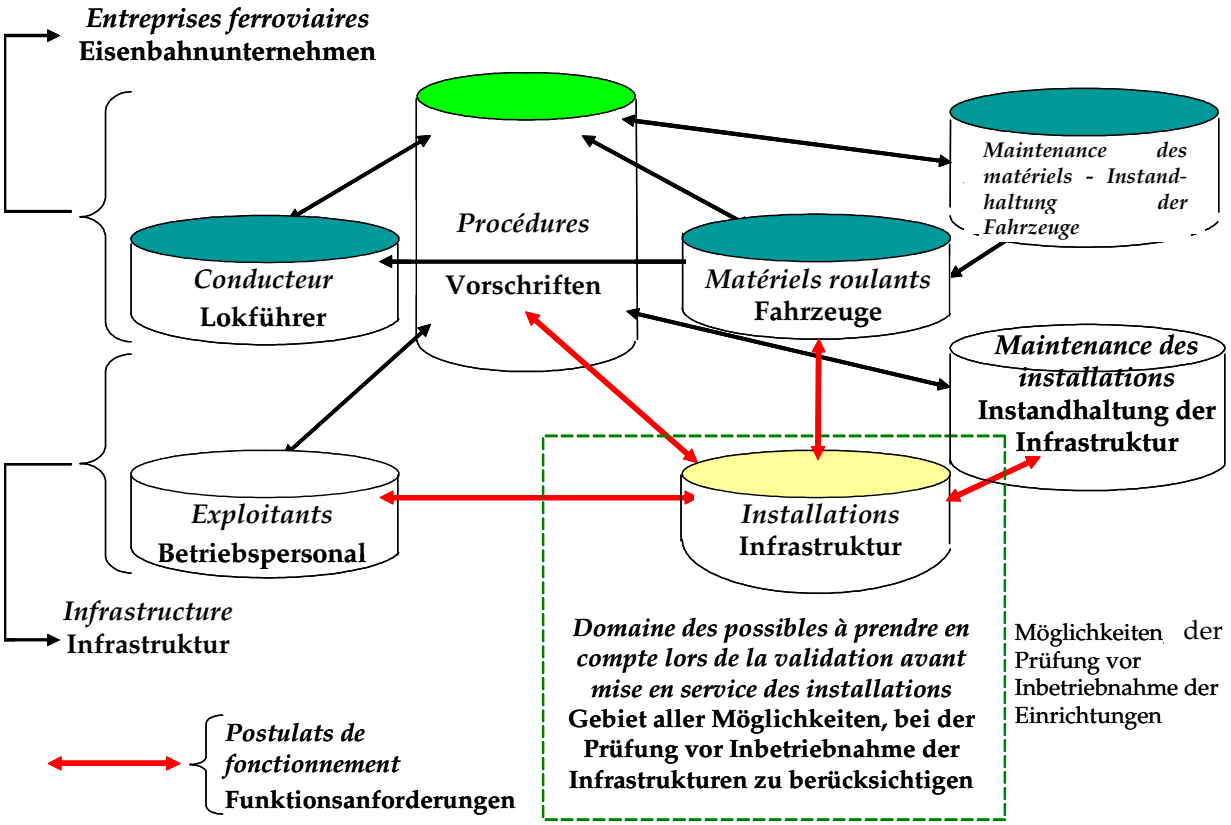



**Figure 4.3** : Vision actuelle globale du système ferroviaire

**Abbildung 4.3**: Heutige Gesamtsicht des Bahnsystems

La Figure 4.3 illustre le fonctionnement en sécurité d'une installation (son aptitude à remplir les différentes fonctions attendues avec le bon niveau de sécurité). Sur chacun des axes il existe des modes dégradés et des possibilités de reprise en manuel.

Abbildung 4.3 zeigt das sichere Funktionieren einer Anlage (seine Fähigkeit, die verschiedenen erwarteten Funktionen mit der entsprechenden Sicherheit zu erfüllen). Auf jeder Achse gibt es Rückfallebenen und manuelle Übernahmen.

Le système ferroviaire se compose donc de 4 sous systèmes (cf. EN RL49/2004) mettant en jeu 4 types d'acteurs. Les procédures cohérentes (postulats de fonctionnement) garantissent la parfaite compréhension des dialogues et la cohérence des actions. Elles constituent le pivot de la sécurité du système. Elles sont d'une part, le produit à la fois d'analyse et de compromis techniques et, d'autre part, de l'expérience accumulée en 150 ans de fonctionnement.

### 4.3.1 Les hommes

L'homme est le maillon incontournable du système de sécurité et la fiabilité du système repose largement autant sur la fiabilité humaine que sur la fiabilité technique [SNCF, 1995] [Reason, 1993] [Bouvarel, 2003] [Poncet, 2008] [Hollnagel, 2004] [Hollnagel, 2007] [Hollnagel, 2008].

Si les accidents ont montré les risques que l'homme a de se tromper, le retour d'expérience montre qu'il dispose de facultés de synthèse et d'anticipation lui permettant de s'adapter à des situations imprévues, à intégrer des facteurs multiples et divers et de rattraper des situations délicates.

Les systèmes très automatisés, très élaborés et très fiables posent le redoutable problème de la surveillance de ces outils par l'homme et des effets de la résilience. En effet, les probabilités de dysfonctionnement contraires à la sécurité sont si faibles que la plupart des opérateurs ne les rencontreront pas fréquemment dans leur vie. Or, certains de ces dysfonctionnements réputés «sûrs» nécessitent de pouvoir compter sur les opérateurs dans le cadre l'application de procédure (malheureusement avec un taux d'erreur non négligeable).

L'homme est le seul capable de faire face à toutes les situations pour l'exploitation des outils et notamment en cas de défaillance et de dérangement qui font généralement appel à l'utilisation de procédures et ce avec un taux de vigilance réduit par rapport à la situation sans installation de sécurité [SNCF, 1995] [Hollnagel, 2007] [Poncet, 2008]. On peut le schématiser comme dans la figure 4.4.

Das Bahnsystem umfasst also vier Komponenten (siehe EN RL49/2004) und beinhaltet vier verschiedene Akteure. Kohärente Vorschriften (Anforderungen an die Funktionsweise) gewährleisten das perfekte Verständnis bei der Kommunikation und die Kohärenz der Aktionen. Sie sind das Kernstück der Systemsicherheit. Diese Vorschriften entstanden sowohl aus technischen Analysen als auch aus Kompromissen, sowie aus den 150 Jahren betrieblicher Erfahrung.

### 4.3.1 Der Mensch

Der Mensch ist ein unverzichtbares Glied des Sicherungssystems. Die Zuverlässigkeit des Systems beruht sowohl auf der Zuverlässigkeit des Menschen als auch auf der Zuverlässigkeit der Technik [SNCF, 1995] [Resaon, 1993] [Bouvarel, 2003] [Poncet, 2008] [Hollnagel, 2004] [Hollnagel, 2007] [Hollnagel, 2008].

Einerseits offenbaren Unfälle das Risiko eines menschlichen Versagens, andererseits zeigt die Erfahrung, dass der Mensch zusammenfassend und vorausschauend handeln kann, wodurch er sich an eine unvorhergesehene Situation anpassen, zahlreiche Faktoren berücksichtigen und auch heikle Situationen „retten" kann.

Bei einem weitgehend automatisierten, sehr ausgereiften und sehr zuverlässigen System stellt sich das befürchtete Problem bei der Überwachung durch den Menschen, sowie dem Effekt der Widerstandsfähigkeit. Die Wahrscheinlichkeit einer sicherheitswidrigen Fehlfunktion ist nämlich so gering, dass das Bedienpersonal sie nicht häufig erlebt. Nun ist es jedoch so, dass man bei einigen dieser „sicheren" Fehlfunktionen bei der Anwendung der Vorschriften auf das Bedienpersonal angewiesen ist.

Einzig und alleine der Mensch ist in der Lage, alle betrieblichen Situationen zu meistern (Anwendungsprogramme), insbesondere im Falle eines Fehlers bzw. einer Störung, wo im Allgemeinen auf Vorschriften zurückgegriffen wird. Dies geschieht jedoch mit reduzierter Wachsamkeit im Vergleich zu einer Situation ohne Sicherheitsanlagen [SNCF, 1995] [Hollnagel, 2007] [Poncet, 2008] und kann in Abbildung 4.4 dargestellt werden.

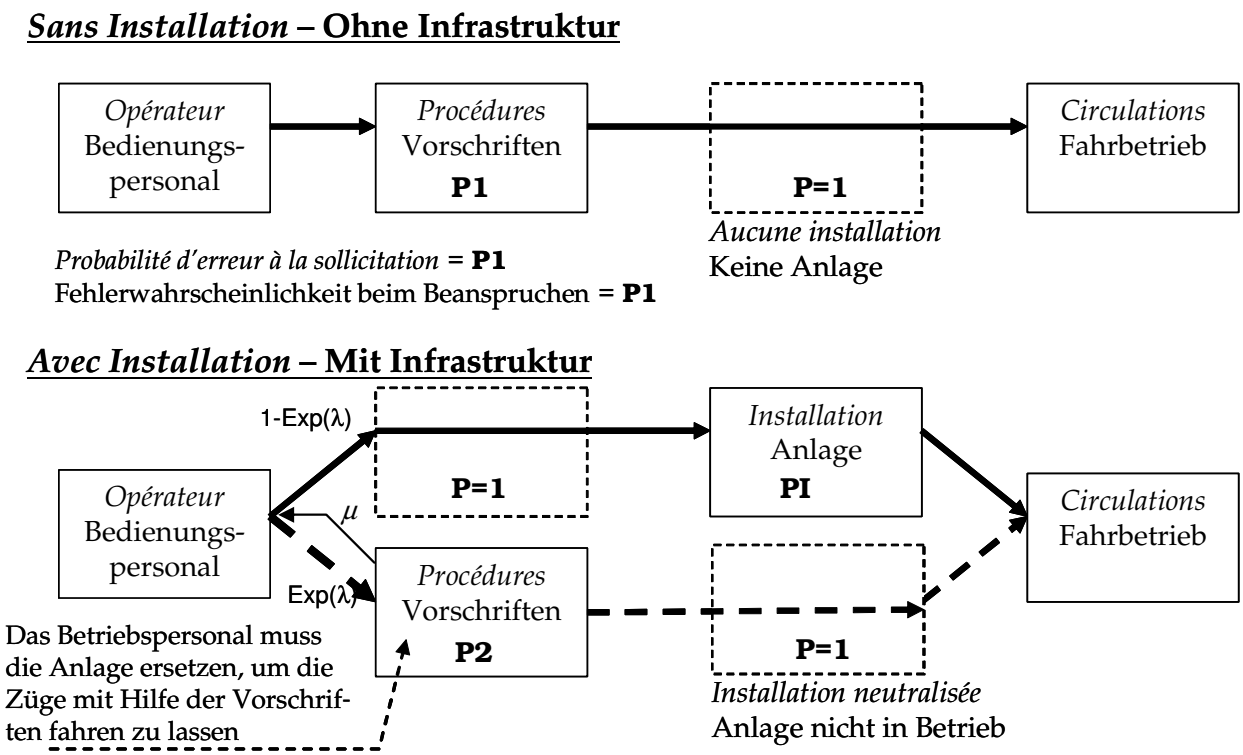



**Figure 4.4** : Effet de résilience dans le traitement des modes dégradé au travers des procédures

**Abbildung 4.4**: Wirkung der Widerstandsfähigkeit bei Bearbeitung der Rückfallebene mithilfe von Vorschriften

Les conditions d'application des procédures par les opérateurs font donc partie intégrante des **postulats de fonctionnement** à prendre en compte pour toute validation d'un système de sécurité tel qu'un poste d'aiguillage. L'occurrence de blocages ou pannes sures des installations conduit à une réduction du niveau global de sécurité.

**Il est nécessaire de détecter les éventuels surabondants qui peuvent de manière assez paradoxale réduire le niveau de sécurité global d'un ensemble «Opérateur–Procédures-Installation »** [Hollnagel, 2007] [Poncet, 2008].

L'homme, dans le cadre des procédures, joue le rôle principal dans le système ferroviaire.

La fiabilité des installations à sa disposition induit naturellement très vite chez l'opérateur une tendance lourde à lui faire confiance (tunnel mental), et à écarter tout signe qui ne confirmerait pas la bonne marche de l'installation.

Die Anwendungsbedingungen der Vorschriften durch das Personal gehören somit voll und ganz zu den **Funktionsforderungen**, die bei jeder Überprüfung eines Sicherungssystems (beispielsweise eines Stellwerkes) zu berücksichtigen sind. Das Vorkommen einer Blockade oder einer „sicheren" Störung der Anlage führt zu einer Reduzierung des Gesamtsicherheitslevels.

**Etwaige „überflüssige" Forderungen (Anforderungen, die sich ggf. widersprechen und die deswegen zu einem sicheren, aber unerwüschten Zustand führen, der unter unvorhersehbaren Bedingungen zu einem unsicheren Zustand werden kann), die auf ziemlich paradoxe Weise den Gesamtsicherheitslevel der Einheit „Betreiber - Vorschrift - Anlage" reduzieren können, müssen entdeckt werden** [Hollnagel, 2007] [Poncet, 2008].

Im Bezug auf die Vorschriften spielt der Mensch beim Bahnsystem die Hauptrolle.

Die hohe Zuverlässigkeit der Anlagen, über die das Betriebspersonal verfügt, führt auf natürliche Weise sehr schnell dazu, dass das Personal stark dazu neigt, den Anlagen zu vertrauen (mentaler Tunnel) und das nicht vorschriftsmäßige Funktionieren der Anlage zu übersehen.

## 4.3.2 Les procédures

Les procédures sont conçues pour assurer la cohérence des actions des différents acteurs entre eux et avec le fonctionnement des installations et des équipements. La première des procédures vise à unifier le vocabulaire pour que les termes utilisés aient le même sens pour tous les opérateurs.

Les procédures ferroviaires doivent être mises en œuvre par des opérateurs généralement isolés. Elles visent donc à codifier la conduite à tenir dans l'ensemble des situations (surtout dégradées et/ou perturbées), avec tous les types de matériel roulant et sur toutes les infrastructures.

Sur les systèmes informatiques, très automatisés, la procédure se réduit aux cas de dérangement des installations et son application est peu courante et donc délicate. Il est à noter qu'elle est essentielle pour permettre avec des performances dégradées d'écouler en sécurité un minimum de trafic et, a minima, d'évacuer les trains prisonniers.

Exemples de procédure :

- L'agent de conduite doit une obéissance passive et immédiate aux signaux le concernant. En cas d'extinction d'un signal, l'agent de conduite doit adapter sa conduite comme si le signal présentait l'indication la plus restrictive qu'il peut présenter ;
- L'aiguilleur peut annuler l'effet de l'enclenchement des aiguilles lorsqu'il a l'assurance de la «non libération» infondée d'un ou plusieurs circuit de voie afin de faire manœuvrer en mode dégradé une aiguille et ainsi faire circuler une circulation ferroviaire. Il est clair que même s'ils en ont la possibilité à tout moment, les opérateurs ne doivent utiliser ces possibilités qu'en cas de besoin et dans les conditions prescrites ;
- L'implantation des signaux carrés est telle qu'une distance de glissement est réservée entre le signal et le point à protéger. Cette distance peut sembler importante, mais elle ne représente que 6% environ de la distance d'arrêt totale[21].

## 4.3.2 Vorschriften (Betriebsregeln)

Die Vorschriften sichern die Kohärenz der Aktionen verschiedener Akteure untereinander sowie ihre Interaktionen mit der Funktionsweise der Anlagen und Einrichtungen. Die erste Vorschrift betrifft die Vereinheitlichung der Wörter, damit die verwendeten Termini für das gesamte Betriebspersonal dasselbe bedeuten.

Die Vorschriften des Bahnverkehrs müssen im Allgemeinen von auf verschiedene Orte verteiltem Bedienpersonal umgesetzt werden. Es geht also darum, die Verhaltensweisen für alle Situationen (vor allem für die Rückfallebene und/oder die gestörte Situation), für alle Fahrzeugtypen und für alle Infrastrukturarten festzulegen.

Auf weitgehend automatisierten Rechnersystemen beschränken sich die Vorschriften auf Anlagestörungen und ihre Anwendung ist selten und daher heikel. Es ist anzumerken, dass die Vorschriften sehr wichtig sind, damit bei verschlechterter Kapazität ein Minimum an Verkehr in aller Sicherheit gewährleistet werden kann und die liegengebliebenen Züge zumindest noch evakuiert werden können.

Beispiele für Vorschriften:

- Der Lokführer muss den ihn betreffenden Signalen sofort passiv gehorchen. Falls ein Signal erloschen ist, muss der Lokführer seine Fahrweise derart anpassen, als hätte er das restriktivste Signal vor sich.
- Der Weichensteller kann den Weichenverschluss aufheben, falls er die Gewissheit hat, dass ein oder mehrere Gleisstromkreise unbegründet belegt sind (Gleis nicht frei), damit die Weiche in der Rückfallebene bedient werden und der Bahnbetrieb auf diese Weise weitergehen kann. Es ist klar, dass die Weichensteller jederzeit diese Möglichkeit haben, sie jedoch nur im Bedarfsfall nutzen dürfen unter den vorgeschriebenen Bedingungen
- Der Standort der Hauptsignale ist so festgelegt, dass ein Durchrutschweg zwischen dem Signal und dem zu schützenden Punkt vorhanden ist. Dieser Abstand mag lang scheinen, aber er stellt nur ungefähr 6 % des vollen Bremswegs dar[22].

[21] Si l'on y ajoute l'incertitude quant à l'estimation, par le conducteur, du moment auquel il doit commencer le freinage, de l'effort et de sa modulation, la précision réelle peut être nettement moins bonne. Il n'est pas exceptionnel qu'un jeune conducteur arrête son train avec une voiture au-delà du quai. Quant aux trains de marchandises, on considère que la précision de leur arrêt est meilleure que… cent mètres

[22] Wenn man dazu die Unsicherheit bei der Einschätzung des Bremszeitpunktes, des Bremswegs und der Steuerung durch den Lokführer hinzufügt, kann die Genauigkeit des wirklichen Haltepunktes noch schlechter sein. Es ist nicht ungewöhnlich, dass ein junger Lokführer seinen Zug mit einem Wagen außerhalb des Bahnsteigs anhält. Bei Güterzügen schätzt man die Bremsgenauigkeit auf besser als 100m.

**L'application des procédures par les opérateurs fait partie des postulats à prendre en compte pour la validation d'un système de sécurité.**

### 4.3.3 Les installations

Les installations se composent des équipements de sécurité à bord des trains et des installations de sécurité au sol. Ils permettent à l'homme de sécuriser son action et d'améliorer ses performances. Ils utilisent des composants de grande fiabilité et très sûrs.

La tendance actuelle est de les transformer en automatismes se substituant progressivement à l'homme et à ses erreurs dans les tâches de sécurité.

Il est traditionnellement distingué :

→ les organes de sécurité des matériels roulants :
- o les organes de roulement dont la pérennité de la tenue mécanique doit être assurée dans toutes les conditions de fonctionnement ;
- o le frein qui doit être disponible y compris lors des freinages prolongés sur fortes pentes ;
- o le dispositif de veille automatique avec contrôle du maintien d'appui qui est destiné à contrôler l'état de veille du conducteur en rendant impossible le blocage du dispositif,
- o les équipements de contrôle de vitesse ;
- o les équipements de signalisation embarquée (signalisation de cabine).

→ les installations de sécurité au sol constituées par :
- o les signaux qui permettent de transmettre leurs ordres aux conducteurs ;
- o les appareils de voie qui permettent les changements de direction ;
- o les enclenchements qui assurent la sécurité des circulations sur les appareils de voie ;
- o les passages à niveau (points délicats du système ferroviaire).

Il est à noter que par rapport à ses voisins européens, la SNCF a toujours considéré que l'agent de conduite avait une certaine valeur d'attention, et qu'il ne fallait pas en perdre le bénéfice [Bouvarel, 2003] [Poncet, 2008].

**Die Anwendung der Vorschriften durch das Bedienungspersonal gehört zu den für die Freigabe eines Sicherungssystems zu berücksichtigenden Anforderungen.**

### 4.3.3 Anlagen

Die Anlagen umfassen die Sicherungseinrichtungen an Bord der Züge sowie die Sicherungsanlagen im Gleis. Mit ihrer Hilfe kann der Mensch sein Handeln absichern und seine Leistung verbessern. Es werden hierfür Bauteile mit großer Zuverlässigkeit und großer Sicherheit benutzt.

Der heutige Trend ist die Umwandlung der Anlagen in Automatismen, die den Menschen und seine Fehler bei Sicherungsaufgaben nach und nach ersetzen sollen.

Üblicherweise wird folgendes unterschieden:

→ Sicherheitsbauteile der Fahrzeuge:
- o Fahrwerke, deren mechanischer Halt unter allen Funktionsweisen gewährleistet sein muss
- o Bremsen, die verfügbar sein müssen, auch bei länger andauernden Bremsungen auf starkem Gefälle
- o Sicherheitsfahrschaltung mit Betätigungskontrolle zur Überprüfung der Wachsamkeit des Fahrzeugführers, welche nicht blockiert oder ausgeschaltet werden kann
- o Geschwindigkeitskontrolleinrichtungen
- o Signalapparatur (Signalsystem im Führerraum).

→ Sicherungsanlagen im Gleis umfassen:
- o Signale zur Übertragung der Befehle an die Fahrzeugführer
- o Weichen, die den Fahrtrichtungswechsel erlauben
- o Verschlüsse, die die Fahrsicherheit auf den Weichen sichern
- o Bahnübergänge (gefährliche Bereiche des Bahnsystems).

Es ist anzumerken, dass die SNCF - im Vergleich zu ihren europäischen Nachbarn - immer davon ausging, dass der Fahrzeugführer aufmerksam ist, und dass dieser Vorteil nicht verloren gehen darf [Bouvarel, 2003] [Poncet, 2008].

Ainsi la TVM a été conçue avec des «visualisateurs de sécurité» correspondant à un niveau de confiance extrêmement élevé et un contrôle de vitesse seulement « sûr ». En utilisant les qualités du conducteur ($10^{-3}$ - $10^{-4}$/h environ), le contrôle de vitesse n'a pas besoin de dépasser un taux de mise en défaut tel que son produit par $10^{-3}$ ou $10^{-4}$ atteigne les objectifs globaux.

Les futurs systèmes européens exigent un contrôle de vitesse « de sécurité » qui « porte » alors toute la sécurité. Ce qui impose de retenir des décélérations garanties de freinage très inférieures aux performances réelles et donc de réduire le débit et donc les performances apparentes du système ferroviaire.

Ceci ne sera pas sans impact sur les coûts de maintenance des matériels roulants afin de « maintenir dans le temps » ces « décélérations garanties ».

## 4.4 Les risques couverts par la signalisation

### 4.4.1 Les événements dangereux et propriétés de sécurité

Les caractéristiques du chemin de fer conduisent à énoncer les règles fondamentales du tableau 4.1 :

| Caractéristiques | | Règles fondamentales |
|---|---|---|
| un système à une seule dimension | ⇔ | deux trains ne peuvent se trouver au même endroit au même moment |
| adhérence et frottement limités | ⇔ | un train doit disposer d'une distance libre au moins égale à sa distance de freinage |
| point de géométrie de plus d'une dimension | ⇔ | un train ne peut s'engager qu'avec l'assurance de non déformabilité et de présence d'autre train |

**Tableau 4.1** : Règles fondamentales du système ferroviaire

De l'application de ces règles, il en résulte 5 types de configurations, ou encore 7 types d'évènements dangereux que l'on eut les associer selon le tableau 4.2.
A chaque risque correspond un ensemble de propriétés de sécurité devant être respectées en permanence par l'un ou l'autre des composantes du système ferroviaire, notamment les installations de sécurité au sol (signalisation et poste d'aiguillage) et les matériels roulants.

Aus diesem Grund ist das Signalsystem im Führerstand (TVM) mit „Sicherheitsanzeigen" ausgestattet, die einen extrem hohen Sicherheitslevel vorweisen; die Geschwindigkeitskontrolle ist hingegen weniger sicher. Durch die Einbeziehung des Fahrzeugführers (Sicherheitslevel zwischen $10^{-3}$ und $10^{-4}$/h) ist es nicht notwendig, dass die Geschwindigkeitskontrolle sicherer ist als eine Fehlerrate die mit $10^{-3}$ oder $10^{-4}$ multipliziert das Zielniveau erreicht.

Die zukünftigen europäischen Systeme fordern eine „Sicherheitsgeschwindigkeitskontrolle", die alleine die gesamte Sicherheit realisiert: dafür muss die gewährleistete Bremsverzögerung sehr viel niedriger sein als die tatsächliche Leistung, so dass die Streckenauslastung und so die tatsächliche Leistung des Bahnsystems reduziert werden.

Dies hat einen Einfluss auf die Instandhaltungskosten der Fahrzeuge, die diese garantierten Bremsverzögerungen längerfristig gewährleisten müssen.

## 4.4 Durch Signaltechnik abgedeckte Gefährdungen

### 4.4.1 Gefährliche Ereignisse und Sicherheitseigenschaften

Die Eigenschaften des Bahnsystems führen zu den grundlegenden Regeln der Tabelle 4.1.

| Eigenschaften | | Grundregeln |
|---|---|---|
| Eindimensionales System | ⇔ | zwei Züge können nicht gleichzeitig an derselben Stelle sein |
| Beschränkte Haftreibung / Reibung | ⇔ | ein Zug muss über einen freien Fahrweg verfügen, der mindestens seinem Bremsweg entspricht |
| Stellen mit mehr als einer Dimension | ⇔ | ein Zug kann nur dann diese Stelle passieren, wenn sicher ist, dass sie unverändert bleibt und kein anderen Zug vorhanden ist |

**Tabelle n°4.1**: Grundregeln des Bahnsystems

Durch die Anwendung dieser Regeln ergeben sich fünf Konfigurationen oder sieben Arten von Gefährdungen Ereignissen gemäß Tabelle 4.2.

Zu jedem Risiko gehört eine bestimmte Anzahl von Sicherheitseigenschaften, die ständig von einer der Komponenten des Bahnsystems eingehalten werden müssen, insbesondere von den Sicherheitsanlagen im Gleis (Signalsystem und Stellwerk) und den Fahrzeugen.

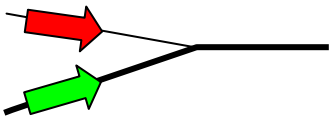
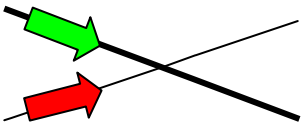
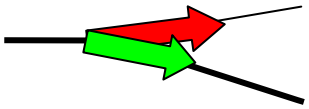
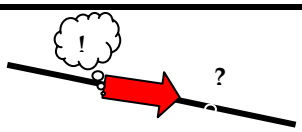
| Configurations<br>Konfigurationen | Événements dangereux<br>Gefährliche Ereignisse | | |
|---|---|---|---|
| **1° Affrontement / Konfrontation** | **1° Nez à nez / Frontalzusammenstoß** | *Deux circulations se dirigeant simultanément l'une vers l'autre* / Zwei Züge, die gleichzeitig aufeinander zufahren |  |
| **2° Convergence & Cisaillement / Zusammenlaufen und Kreuzen der Fahrwege** | **2° Prise en écharpe / Flankenfahrt** | *Deux circulations se dirigeant simultanément vers une même voie* / Zwei Züge, die gleichzeitig auf dasselbe Gleis zufahren |  |
| | | *Deux circulations empruntant simultanément des itinéraires sécants* / Zwei Züge, die gleichzeitig in sich kreuzende Fahrstraßen fahren |  |
| **3° Divergence / Auseinanderlaufen** | **3°Entrebâillement / Zungenklaffen** | *Mouvement d'un aiguillage sous une circulation* / Weichenbewegung unter einem fahrenden Zug |  |
| **4° Succession / Zugfolge** | **4° Rattrapage / Auffahren (Einholen)** | *Deux circulations évoluant dans le même sens se rapprochant ou se rattrapant* / Zwei Züge in derselben Richtung, die einander näher kommen oder sich einholen |  |
| **5° Guidage / Spurführung** | **5° Dérive / Wegrollen** | *Circulation sans frein (ne respectant donc pas la signalisation d'arrêt ou de limitation de vitesse)* / Zug ohne Bremse (der also Haltsignale und Geschwindigkeitsbegrenzungen ignoriert) |  |
| | **6° Déraillement / Entgleisung** | *Circulation perdant le guidage des rails du fait d'une survitesse ou d'un défaut de voie* / Zug der aufgrund einer erhöhten Geschwindigkeit oder eines Gleisfehlers die Spurführung verliert |  |
| | **7° Collision / Zusammenstoß** | *Circulation heurtant un obstacle non annoncé (obstacle naturel ou au passage à niveau)* / Zug der mit einem ungemeldeten Hindernis zusammenstößt (natürliches Hindernis oder Bahnübergang) |  |

**Tableau n°4.2** : Configurations et risques et événement redoutés du système ferroviaire
**Tabelle n°4.2**: Konfigurationen, Risiken und befürchtete Ereignisse des Bahnsystems

## 4.4.2 Le nez à nez

Sur une ligne à une seule voie, les trains ne peuvent se croiser que dans les gares. Entre les gares (en ligne), l'événement dangereux possible est donc le nez à nez :

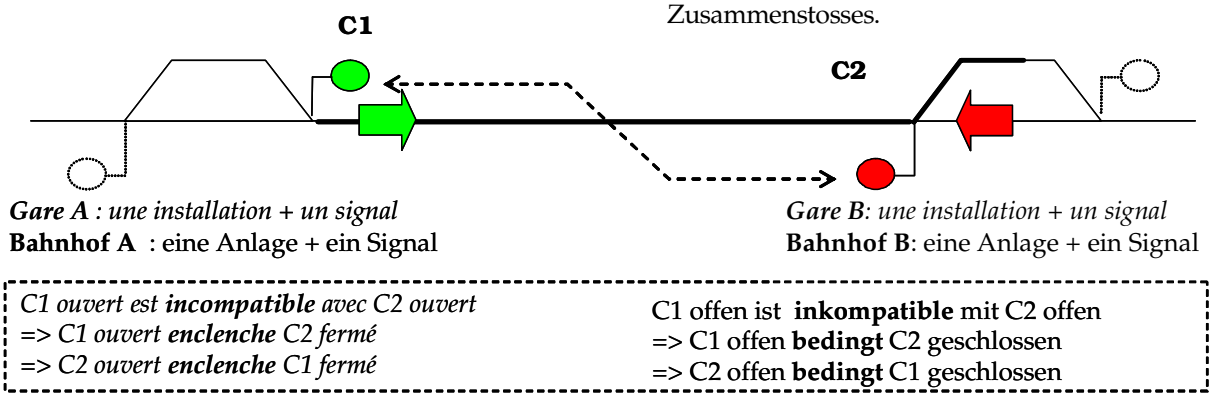



**Figure 4.5** : Enclenchement de Nez à Nez

De plus le train doit respecter une limitation de vitesse à l'entrée de la gare B sous peine de dérailler sur l'aiguille d'entrée.
Dans certains cas la sécurité des circulations repose encore uniquement sur les procédures et les hommes (procédures particulières propres à la «voie unique»), ce qui n'est possible que lorsque les gares sont occupées par des opérateurs et que le trafic reste faible (moins d'une dizaine de circulations par jour).

Lorsque le trafic s'élève, les installations de sécurité, notamment celles informatiques, doivent assurer la protection des mouvements en affrontement. L'opérateur commande alors à distance les installations pour ordonnancer les circulations et, le cas échéant, n'applique les procédures qu'en cas de dysfonctionnement des installations ce qui requiert des moyens de communication avec les circulations.

**L'enclenchement dit de «voie unique» traduit les propriétés de sécurité couvrant le risque de nez à nez involontaire entre deux circulations partageant la même voie.**

## 4.4.2 Frontalzusammenstoß

Auf einer eingleisigen Strecke können sich die Züge nur in den Bahnhöfen kreuzen. Zwischen den Bahnhöfen (auf der Strecke) besteht, wie in Abb. 4.5 gezeigt, die Gefahr eines frontalen Zusammenstosses.

**Abbildung 4.5**: Verschluss/Fahrstraßenbildung zur Verhinderung eines Frontalzusammentoßes

Der Zug muss am Eintritt des Bahnhofs B langsam fahren sonst wird er auf der Eintrittweiche entgleisen.
In einigen Fällen beruht die Fahrsicherheit lediglich auf den Vorschriften und auf den Menschen (besondere Verfahren für eingleisige Strecken), was nur dann möglich ist, wenn in den Bahnhöfen Betriebspersonal vorhanden ist und wenn das Verkehrsaufkommen niedrig ist (weniger als zehn Züge pro Tag).

Wenn das Verkehrsaufkommen ansteigt, müssen Sicherheitsanlagen, insbesondere IT-Anlagen, die Sicherheit der entgegengesetzten Zugbewegungen gewährleisten. Das Bedienungspersonal steuert die Anlagen aus der Ferne, um die Zugfolge festzulegen. Gegebenenfalls werden Vorschriften angewandt aber nur im Falle einer Fehlfunktion der Anlagen. Dies erfordert Kommunikationsmöglichkeiten mit den Zügen.

**Der Verschluss auf einer eingleisigen Strecke setzt die Sicherheitseigenschaften um, die nötig sind, um das Risiko eines Frontalzusammenstoßes zwischen zwei Zügen, die sich dasselbe Gleis teilen, zu verhindern.**

## 4.4.3 La prise en écharpe

La prévention de la prise en écharpe est assurée par les installations de sécurité commandées par une installation particulière appelée poste d'aiguillage qui assure:

- la présentation d'un signal d'arrêt absolu (signal de protection) sur les autres itinéraires ;
- la mise en protection des aiguilles pouvant donner accès à un itinéraire tracé ;
- la présence d'impasse de sécurité à la jonction des voies de service et des voies principales.

## 4.4.3 Flankenfahrt

Flankenfahrten wird durch bestimmte Sicherheitsanlagen, den Stellwerken, vorgebeugt:

- Anzeigen eines absoluten Haltbefehls (Zugdeckungssignal) auf allen anderen Fahrstraßen
- Schutz der Weichen, die Zugang zu einer schon eingestellten Fahrstraße geben könnten
- Schutz an der Verbindung der Nebengleise mit den Hauptgleisen.

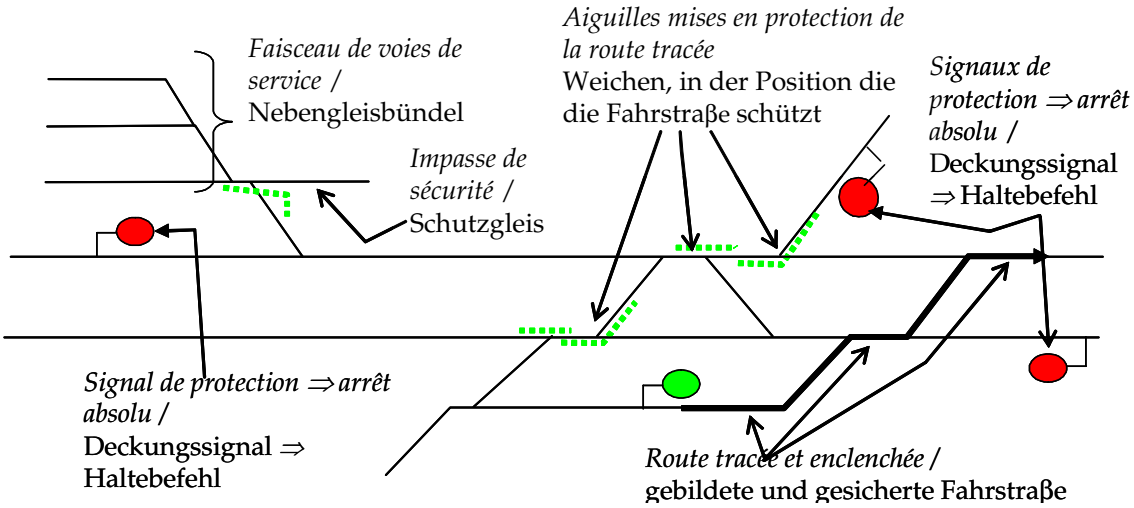



**Figure 4.6** : Enclenchement de formation d'itinéraire couvrant les prises en écharpe

**Abbildung 4.6**: Fahrstraßenbildung, die die Flankenfahrt verhindert

**L'enclenchement dit de « formation d'itinéraire » traduit les propriétés de sécurité couvrant les risques de convergence et de nez à nez entre deux circulations traversant la même zone d'action du poste d'aiguillage.**

**Der Fahrstraßenbildungsverschluss setzt die Sicherheitseigenschaften zur Vermeidung des Risikos eines Zusammenlaufens der Fahrwege und eines Frontalzusammenstoßes zwischen zwei Zügen, die sich im gleichen Stellwerksbereich befinden, um**.

## 4.4.4 L'entrebâillement

Il concerne les aiguilles : parties mobiles des appareils de voie. La prévention est traitée par les installations de sécurité qui interdisent la translation en présence d'une circulation (calage du mécanisme de manœuvre et verrouillage des lames mobiles) et le contrôle permanent de la position correcte des parties mobiles.

## 4.4.4 Zungenklaffen

Das Zungenklaffen betrifft die beweglichen Teile der Weichen. Es wird durch Sicherheitsanlagen verhindert, die im Falle der Präsenz eines Zuges, das Umlegen der Weiche verbieten (Verkeilung der Weichenbedienungsantriebe und Verschluss der beweglichen Zungen) und die korrekte Lage der beweglichen Teile der Weiche ständig kontrollieren.

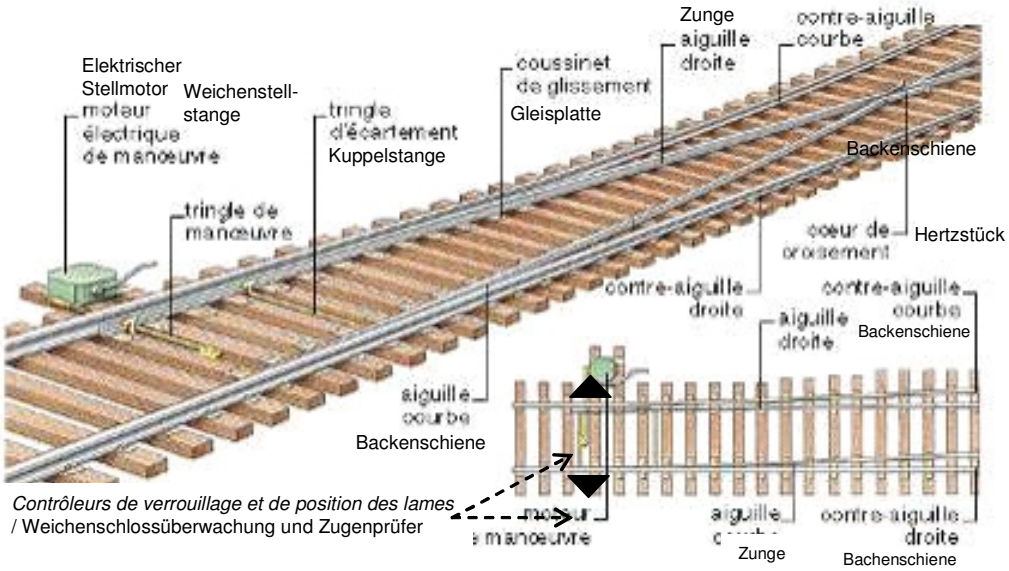



**Figure 4.7** : Enclenchement d'établissement d'itinéraire couvrant les risques d'entrebâillements

**Abbildung 4.7**: Fahrstraßenbildung zur Verhinderung eines Zungenklaffens

**L'enclenchement dit «d'établissement de l'itinéraire» traduit les propriétés de sécurité couvrant les risques d'entrebâillement au niveau des aiguilles parcourues par la circulation sur zone d'action du poste d'aiguillage.**

**Als Sicherung der Fahrstraßenerstellung bezeichnet man die Sicherungseinrichtungen zur Vermeidung eines Zungenklaffens, auf der Ebene der Weichen, so wie im ganzen Stellwerksbereich.**

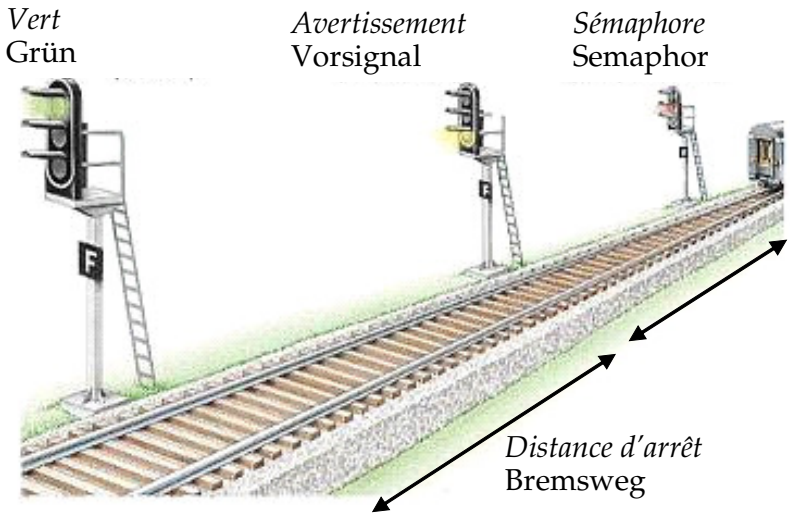



**Figure 4.8** : Espacement automatique (block automatique)

**Abbildung 4.8:** Zugfolgeabschnitte beim selbstätigen Streckenblock

## 4.4.5 Le rattrapage

Pour prévenir le rattrapage, il est nécessaire de mettre en place un système permettant d'espacer les trains et de maintenir entre eux une distance suffisante pour leur permettre de freiner et de s'arrêter au besoin avant la queue du train précédent, même en cas d'arrêt brutal de celui-ci (pour une cause extérieure). Le principe retenu est le cantonnement (c'est-à-dire : la division de la ligne en sections appelées « cantons ») et sur l'interdiction d'autoriser la présence en vitesse de deux circulations[23] dans un même canton. (Fig. 4.8)

Elle suppose une liaison entre le point d'entrée et le point de sortie des cantons pour s'informer mutuellement des circulations. Les installations et la procédure de fonctionnement du cantonnement sont appelées « block ». Les principes de fonctionnement du block entre deux points A et B sont :

- protection du train par le signal d'entrée du canton en A ;
- reddition par B de la voie libre en A que lorsque le train a entièrement dégagé le canton A-B et est arrivé complet en B (n'a pas perdu un de ses véhicules).

En double voie, les trains circulant dans le même sens, il est envisageable d'autoriser les trains à pénétrer en canton occupé. C'est la solution retenue en France, où les blocks automatiques à cantons courts, y compris sur LGV, sont permissifs.

L'agent de conduite pénètre alors de lui-même sans avis d'un opérateur en canton occupé en marquant un arrêt devant le signal d'entrée du canton et en marchant à vue sans dépasser 30km/h sur tout le canton jusqu'au dégagement du signal suivant par le dernier essieu de son train. Cette dernière règle permet de maintenir un débit résiduel sur incident mais nécessite, de distinguer l'aspect des signaux d'arrêt de block des signaux de protection des nez à nez et de protection des prises en écharpe et d'entrebâillement.

## 4.4.5 Auffahren

Um dem Auffahren vorzubeugen, muss ein System installiert werden, das die Züge auf Abstand halten kann und den Bremsweg gewährleistet, den ein Zug benötigt, um vor dem Ende des vorausfahrenden Zuges anhalten zu können selbst bei einer Notbremsung aufgrund einer nichtvorhersehbaren Ursache. Das Prinzip beruht auf einer Einteilung der Strecke in Abschnitte (Blockabschnitte) und auf dem Verbot der Anwesenheit zweier Züge[24] auf demselben Blockabschnitt (Abb. 4.8).

Um die Position eines Zuges zu bestimmen ist also eine Verbindung notwendig zwischen der Einfahrt und der Ausfahrt aus dem Blockabschnitt. Die Anlagen und das Verfahren der Einteilung in Streckenabschnitte heißt „Streckenblock" bzw. „Blocksicherung". Die Funktionsweise der Zugfolgesicherung zwischen zwei Punkten A und B ist wie folgt:

- Schutz des Zuges durch das Eingangssignal des Blockabschnittes in A
- Rückmeldung (durch B) der „Fahrt frei in A", lediglich dann, wenn der Zug den Blockabschnitt zwischen A und B verlassen hat und vollständig in B angekommen ist (der Zug hat keines seiner Fahrzeuge verloren).

Im Falle eines Doppelgleises, auf dem Züge in dieselbe Richtung fahren, kann man in Erwägung ziehen, Züge auf einen besetzten Blockabschnitt einfahren zu lassen. Diese Lösung wird in Frankreich angewandt, wo kurze, automatische Blöcke, auch auf der Hochgeschwindigkeitsstrecke (LGV) permissive Blöcke sind.

Der Fahrzeugführer fährt von selbst, ohne Anordnung des Bedienungspersonals, in den besetzten Blockabschnitt ein. Er muss dazu vor dem Eingangssignal des Blocks anhalten und dann „auf Sicht" (unter 30 km/h) den ganzen Block durchfahren bis das nachfolgende Signal durch das Räumen des Blocks durch den letzten Radsatz des Zuges freigegeben wird. Diese Regel erlaubt es, nach einem Vorfall einen minimalen Durchfluss aufrecht zu erhalten. Sie erfordert jedoch die Unterscheidung zwischen bedingten Blockhaltesignalen und Schutzsignalen (zur Verhinderung von Frontalzusammenstößen, Flankenfahrten oder Zungenklaffen).

[23] Block non permissif ou absolu

[24] Dies gilt für nicht permissive oder absolute Blöcke.

Au contraire, le block est dit **absolu** lorsque que tout franchissement d'un signal fermé impose la délivrance systématique d'une autorisation de l'opérateur sédentaire (cas général en Allemagne).

**L'enclenchement dit de «block» traduit les propriétés de sécurité couvrant les risques de rattrapage d'une circulation par une circulation de même sens mais plus rapide.** [25]

## 4.4.6 Les dérives

La dérive est la mise en mouvement non maîtrisée d'un véhicule. Elle résulte d'un accostage, du vent, de la pente, de la défaillance des freins, d'une mauvaise immobilisation d'un véhicule en stationnement… Elle est favorisée par le faible coefficient de frottement acier sur acier et nécessite des réactions extrêmement rapides.

Pour prévenir cet événement dangereux, les gares sont établies en palier (à niveau) et, lorsque le plan de voie le permet, certaines aiguilles non parcourues sont commandées en position de protection à l'occasion de la formation d'un itinéraire (cf. la figure°4.6).

**L'enclenchement dit de « formation d'itinéraire » traduit les propriétés de sécurité couvrant les risques de dérive non commandée d'une circulation dans une zone d'action affectée à un itinéraire.**

Andererseits gilt der Block als **absolut** (mit unbedingten Haltsignalen), wenn jedes Überfahren eines geschlossenen Signals systematisch eine Erlaubnis seitens des Weichenstellers erfordert (dies ist in Deutschland der allgemeine Fall).

**Die „Blocksicherung" beschreibt die Sicherungseinrichtungen zur Verhinderung des Auffahrens eines Zuges auf einen anderen Zug, der in dieselbe Richtung aber langsamer fährt** [25]**.**

## 4.4.6 Wegrollen

Das Wegrollen ist eine nicht beherrschte Fahrzeugbewegung. Das Wegrollen wird durch den niedrigen Reibwert von Stahl auf Stahl begünstigt und erfordert extrem schnelles Handeln.

Um ein solches gefährliches Ereignis zu vermeiden, sind die Bahnhöfe gestaffelt (verschiedene Ebenen) und, falls es der Gleisplan erlaubt, werden bestimmte, nicht befahrene Weichen bei der Bildung einer Fahrstraße in Schutzstellung gebracht (siehe Abb. 4.6).

**Der „Fahrstraßenbildungsverschluss" ist eine Sicherungseinrichtung, die ein ungesteuertes Wegrollen eines Zuges in einem von der Fahrstraβe betroffenen Bereichs verhindert.**

[25] Définitions en anglais / Definition in Englich :

**ABSOLUTE BLOCK SYSTEM** « A fixed block system in which a train may enter a block section only after the last train ahead has completely cleared the block section and is protected by a stop signal »

**PERMISSIVE BLOCK** « An automatic block system which is completely controlled by track circuits providing protection against following movements. In which system a train may enter slowly a block section after stop before the stop signal »

**ABSOLUTE SIGNAL** « A signal which must not be passed in stop position without a special permission from the operator »

**AUTOMATIC BLOCK SYSTEM** « A block system in which the signals work automatically. Lines with an automatic block system must be equipped with track clear detection »

## 4.4.7 Le déraillement et la collision

Compte tenu des distances de freinage, les collisions contre les obstacles inopinés constituent un point délicat dans la sécurité ferroviaire. Les mesures de prévention s'organisent autour de la réduction de l'occurrence des événements dangereux inopinés et de la limitation des conséquences en cas de danger.

Le déraillement est la perte du contact roue rail, condition fondamentale de fonctionnement du chemin de fer. Les causes sont souvent multiples : survitesse, stabilité de la voie, organes de roulement… Le déraillement conduisant généralement à la création d'un obstacle sur la voie, les mesures de prévention mises en œuvre visent à faire face simultanément aux déraillements et aux collisions.

Le niveau de risque de collision le plus élevé est celui avec un véhicule routier aux passages à niveau. Pour y pallier les circulations ferroviaires sont conçues pour réduire les risques de soulèvement en cas de choc et les installations de sécurité doivent assurer en toutes circonstances la fermeture des barrières au passage des trains.

## 4.4.7 Entgleisen und Zusammenstoß

Augrund der langen Bremswege stellen die unerwarteten Hindernisse einen heiklen Punkt in der Eisenbahnsicherheit dar. Die Präventivmaβnahmen reduzieren die Anzahl der unerwarteten gefährlichen Ereignisse und begrenzen die Folgen der Ereignisse.

Das Entgleisen ist der Verlust des Rad-Schiene-Kontaktes, sprich der grundlegenden Bedingung des Schienenverkehrs. Die Ursachen sind oft vielfältig: Gleisstabilität, Laufwerke. Da ein Entgleisen im Allgemeinen zu einem Hindernis führt, treffen die vorbeugenden Maßnahmen gleichzeitig das Entgleisen und die Zusammenstöße.

Das höchste Risiko besteht beim Zusammenstoß mit einem Straßenfahrzeug am Bahnübergang. Um diesem vorzubeugen, sind Eisenbahnfahrzeuge so konzipiert, dass sie das Anheben des Fahrzeuges im Falle eines Zusammenstoßes vermeiden. Die Sicherheitsanlagen müssen zudem unter allen Umständen sicherstellen, dass die Schranken bei der Durchfahrt des Zuges geschlossen sind.

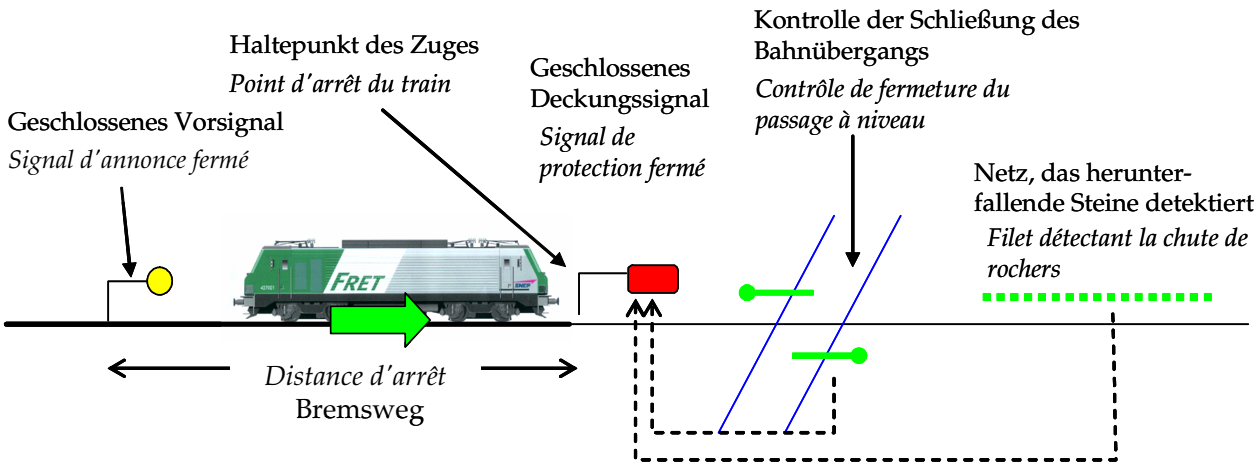



**Figure 4.9** : Enclenchement de contrôle impératif et d'absence de risques identifiés

**Abbildung 4.9**: Permanente Überwachung und Fehlen von identifizierten Risiken

Il est à noter qu'en cas d'obstacle inopiné détecté ou de situation dangereuse en voie, les opérateurs doivent être en mesure d'assurer la protection des circulations dans le cadre de procédures.

**L'enclenchement de « contrôle d'itinéraire » traduit les propriétés de sécurité couvrant les risques de déraillement et de collision pouvant faire l'objet d'un contrôle impératif.**

Es ist anzumerken, dass im Falle eines unerwarteten Hindernisses oder einer gefährlichen Situation auf dem Gleis das Betriebspersonal in der Lage sein muss, im Rahmen der Vorschriften den Schutz der Züge zu sichern.

**Die „Fahrstraßenüberwachung" setzt alle Sicherheitseigenschaften um, die ein Entgleisen oder einen Zusammenstoß verhindern. Dies kann eine zwingende Kontrolle verlangen.**

Certaines portions de voie demandent aux circulations une réduction de la vitesse de franchissement, c'est le cas de certains appareils de voie, ouvrages d'art… Les limitations doivent être annoncées à distance de ralentissement pour les circulations les plus rapides de la ligne.

Bestimmte Gleisabschnitte erfordern eine Geschwindigkeitsreduzierung: das gilt für bestimmte Weichen, Brücken, Tunnel, usw. Die Geschwindigkeitsbegrenzungen müssen früh genug angekündigt werden, damit auch der schnellste Zug seine Geschwindigkeit reduzieren kann.

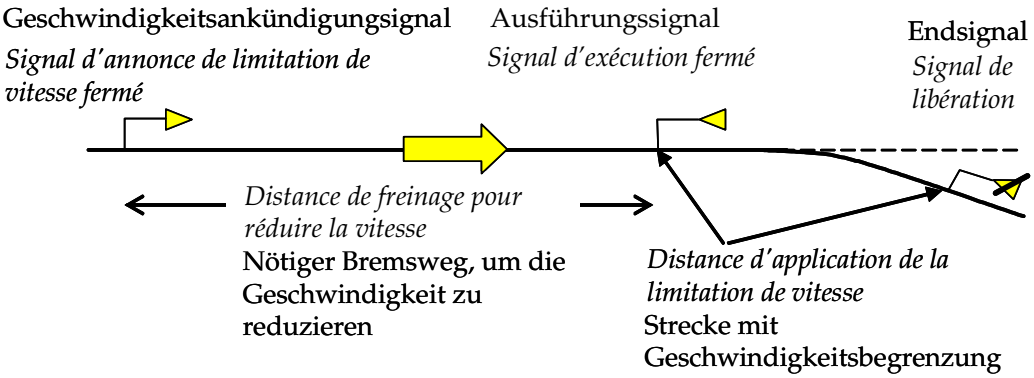



**Figure 4.10** : Signaux de limitation fixes de vitesse sur une portion de voie

**Abbildung 4.10**: Feste Geschwindigkeitsbegrenzung auf einem Gleisabschnitt

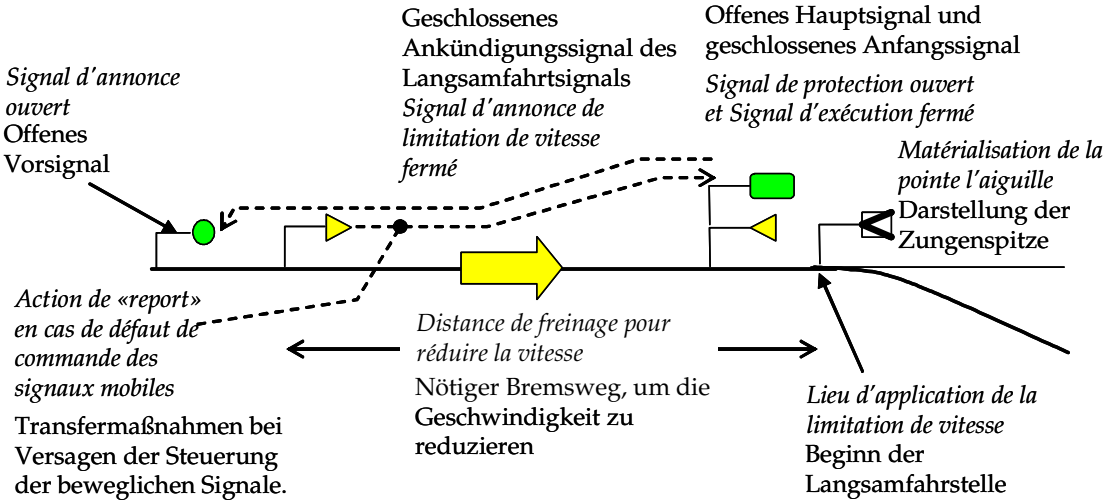



**Figure 4.11** : Signaux de limitation mobile de vitesse sur un appareil de voie (branche déviée)

**Abbildung 4.11**: Verändliche Signale für die Geschwindigkeitsbegrenzung auf einer Weiche (Abzweigung)

**Les enclenchements dits de « report » traduisent les propriétés de sécurité couvrant les risques de mauvaise interprétation par les mécaniciens en cas pannes sûres d'équipements ou de signaux.**

**Die „Rückmeldung" bezeichnet die Sicherungseinrichtung, die im Falle einer sicheren Panne der Gefährdung einer falschen Interpretation der Anlage oder der Signale durch den Fahrzeugführer verhindert.**

Attirons l'attention sur une particularité de la signalisation française (depuis le code VERLANT de 1934) [Annexe A] : les indications lumineuse sont hiérarchisées de manière à ce qu'une seule indication ne soit normalement présentée au mécanicien, toujours avec au maximum trois feux présentés simultanément, (afin de réduire le risque de mauvaise interprétation) en fonctionnement normal de l'installation ou en cas de report sur une indication plus impérative en cas d'extinction d'une indication commandée. A compléter par figure du signal avec hiérarchie des feux. Il existe néanmoins des cas particuliers :

- RR + A[26] : une indication d'annonce et un signal d'exécution de limitation de vitesse ;
- R+RR $\Rightarrow$ RR+A : la même règle conduirait à la présentation de quatre feux, Afin de respecter la règle de présentation d'au plus trois feux lumineux, le R est transformé en A.

Ces règles seront aussi à respecter lors des reports possibles des indications en défaut.

Nachstehende Besonderheit des französischen Signalsystems ist zu beachten (seit dem Code Verlant von 1934) [siehe Anhang A]: aufgrund der Rangfolge der Lichtsignale wird dem Fahrzeugführer bei normaler Funktionsweise der Anlage oder beim Übergang zu einer restriktiveren Anzeige im Fall des Erlöschens eines gesteuerten Signals immer nur eine Anzeige mit höchstens drei Lichtern gleichzeitig präsentiert (zur Reduzierung des Risikos einer falschen Interpretation). Es gibt jedoch Sonderfälle:

- RR + A[27]: ein Vorsignal und der Beginn einer Langsamfahrstrecke
- R+RR $\Rightarrow$ RR+A: dieselbe Regel würde zur Präsentation von vier Lichtern führen. Um die Regel der maximal drei Leuchtbilder einhalten zu können, wird R in A umgewandelt.

Diese Regeln müssen auch bei möglicher Übertragung fehlerhafter Anzeigen anderer Signale eingehalten werden.

[26] Voir Annexe 1

[27] Siehe Anlage 1

# 4.5 Les postes d'aiguillage

## 4.5.1 Introduction

Un poste d'aiguillage (Figure 4.13) est une infrastructure permettant de commander les différents appareils de voie et signaux de protection. Sa mission est la gestion des circulations ferroviaires à l'intérieur d'un espace de voies géographiquement délimité. La gestion de ces circulations consiste à :

- former les itinéraires que les trains doivent emprunter ;
- donner à chaque train présent dans la gare des instructions de mouvement en tenant compte de l'état des itinéraires formés et des positions des autres trains.

Ces actions se décomposent en opérations élémentaires qui doivent être effectuées "en sécurité", c'est à dire dans la certitude d'éviter les collisions et les déraillements [Pichon, 1886] [Descubes, 1898]. Le poste d'aiguillage permet à un opérateur d'effectuer l'ensemble de ces opérations.

# 4.5 Stellwerke

## 4.5.1 Einführung

Ein Stellwerk (Abb. 4.13) ist eine Infrastrukturanlage die Weichen und Zugdeckungssignale steuert. Es regelt den Verkehr innerhalb eines geographisch begrenzten Streckennetzes.
Die Steuerung des Verkehrs umfasst folgendes:

- Bildung der Fahrstraßen, auf denen die Züge fahren sollen
- Jedem im Bahnhof befindlichen Zug sind Zugbewegungsanordnungen zu erteilen - unter Berücksichtigung des Zustandes der gebildeten Fahrstraßen und der Position der anderen Züge.

Diese Aktionen umfassen elementare Vorgänge, die in „Sicherheit" durchgeführt werden müssen, das heißt mit der Gewissheit, Zusammenstöße und Entgleisungen zu verhindern [Pichon, 1886] [Descubes, 1898]. Mit dem Stellwerk kann das Betriebspersonal all diese Vorgänge durchführen.

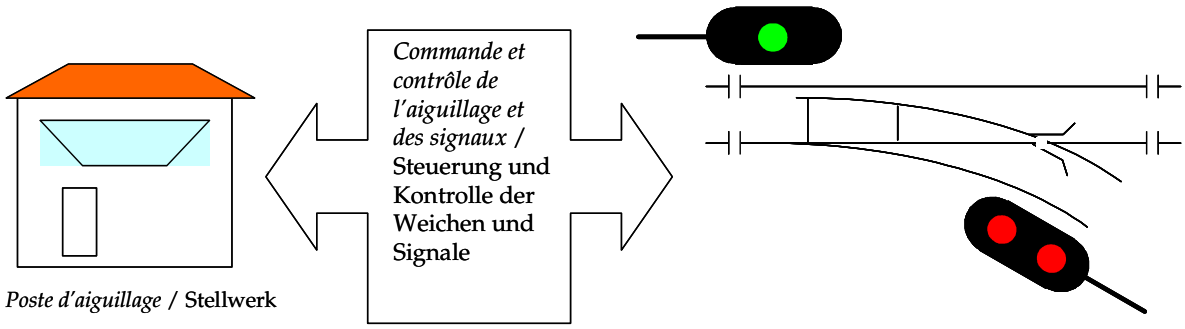



**Figure 4.12** : Poste d'aiguillage

**Abbildung 4.12**: Stellwerk

## 4.5.2 Les phases de fonctionnement d'un poste à itinéraire

Depuis l'avènement des postes d'aiguillage du type PRS, les phases de fonctionnement [Rétiveau, 1987] des postes sont tels que :

- il existe une commande (un bouton ou un message informatique) par itinéraire et un tableau de contrôle optique indépendant ;
- il est possible d'enregistrer un itinéraire avant que ses conditions de formation soient réunies et de disposer d'itinéraires en tracé permanent (certains itinéraires privilégiés restent tracés pour le passage de plusieurs trains qui se succèdent aux conditions du block), de plus son fonctionnement est facilement «lisible» et totalement déterministe ;

## 4.5.2 Funktionelle Phasen eines „Fahrstraßenstellwerks"

Seit dem Aufkommen von Relaisstellwerken mit abschnittsweiser Fahrstraßenauflösung [Rétiveau, 1987], sehen die funktionellen Phasen folgendermaßen aus:

- Es gibt eine Bedienung (Tasten oder Computercode) pro Fahrstraße und eine unabhängige Gleistafel
- Es ist möglich, eine Fahrstraße zu speichern, bevor die Bedingungen zur Fahrstraßenbildung erfüllt sind und eine permanente Stellung der Fahrstraße zu erhalten (bestimmte Fahrstraßen bleiben für die Befahrung durch mehrere, unter den Bedingungen des Blockabschnittes aufeinander folgende Züge gestellt). Die Funktionsweise dieses Stellwerks ist leicht „lesbar" und vollständig deterministisch.

- la réglementation en vigueur précise les modes de traitement des défaillances touchant, le cas échéant, chacune des étapes de la gestion d'un itinéraire.

**Le fonctionnement se fait en phases successives qui tiennent compte des positions successives des appareils de voie et des zones (Tableau 4.3 et Figure 4.13):**

- Das geltende Regelwerk erläutert die Behandlung der Fehler, die in jeder Etappe der Fahrstraßensteuerung auftreten können.

**Der Betrieb läuft in Phasen ab, die die Stellung der Weichen und die Gleisbelegung berücksichtigen (Tabelle 4.3 und Abb. 4.13).** [Retiveau, 1987] [Gernigon, 1998]

| *Phase* **/ Phase** | *Action* **/ Aktion** | *Traduction PRS* **/ Umsetzung durch ein Relaisstellwerk mit abschnittsweiser Fahrstraßenauflösung** | *Résultat* **/ Ergebnis** |
|---|---|---|---|
| **Commande / Steuerung (Anforderung)** | *Enfoncement du bouton de commande* / Drücken der Anforderungstaste | *Fait monter le relais de commande de l'itinéraire (CIt), fait clignoter le bouton de commande* / Betätigt das Anforderungsrelais der Fahrstraße (CIt), lässt die Anforderungstaste blinken | *Ferme les circuits des relais de commande des aiguilles (CAg)* / Schließt die Anforderungsrelais der Weichen (CAg) |
| **Préparation / Vorbereitung** | *Mise en position des aiguilles de l'itinéraire et des aiguilles en protection à condition qu'elles ne soient pas enclenchées par un autre itinéraire (transit non pris) ou occupées par une circulation* / Weichenstellung (Fahrstraße und Schützweichen), falls diese nicht durch eine andere Fahrstraße gesperrt oder durch einen Zug besetzt sind | *Les relais de commande des aiguilles (CAg) basculent à condition que les relais de transit concernés par l'itinéraire soient hauts et les relais de circuit de voie soient libres* / Umschaltung der Weichenanforderungsrelais (CAg), falls die von der Fahrstraße betroffenen Fahrstraßenrelais in der oberen Stellung und die Gleisstromkreisrelais frei sind | *La mise en position de tous les relais de commande d'aiguille (CAg) concernés par l'itinéraire permet le basculement en position d'ouverture du relais d'enclenchement d'itinéraire du signal d'entrée concerné (EIt)* / Sobald alle die Fahrstraße betreffenden Weichenanforderungsrelais in Position gebracht wurden (CAg), kann das Fahrstraßenverschlussrelais des betreffenden Einganssignals (Elt) geöffnet werden |
| **Formation et enclenchement / Fahrstraßenbildung und Sicherung** | *Assure l'immobilisation de l'itinéraire - Interdit la formation d'itinéraires incompatibles* / Sichert die Blockierung der Fahrstraße ab - Verbietet die Bildung inkompatibler Fahrstraßen | *Coupe l'alimentation des relais de transit qui assurent l'enclenchement de transit, fait monter le relais répétiteur d'itinéraire (Rit) qui assure le contrôle permanent de la position des CAg et de EIt* / Unterbricht die Stromversorgung der „Transitrelais" (Fahrstraßenrelais), die den Transitverschluss sichern, bringt das Fahrstraßenwiederholungsrelais (RIt), das die ständige Kontrolle der Stellung der CAg und Elt durchführt, in die obere Stellung | *Fait allumer en fixe le bouton de commande, ferme les circuits de contrôle des itinéraires* / Lässt die Anforderungstaste aufleuchten, schließt die Überwachungskreise der Fahrstraßen |
| **Contrôle (établissement) / Kontrolle** | *Donne l'assurance de la réalisation effective de l'itinéraire et ouvre le signal d'entrée* / Gewährleistet die Realisierung der Fahrstraße und öffnet das Eingangssignal | *Le contrôle impératif des aiguilles est assuré après leur mise en position effective par les relais KAg - Permet de faire monter le relais de contrôle d'itinéraire du signal d'entrée Kit C1* / Die verbindliche Überprüfung der Weichen erfolgt nach deren tatsächlichen Stellung durch die KAg (Weichenüberwachungs-) Relais - Dadurch wird das Fahrstraßenüberwachungsrelais des Eingangssignals KIt Cl in die obere Stellung gebracht | *Ouverture du signal d'entrée Extinction du voyant de contrôle de fermeture du signal d'entrée* / Öffnen des Eingangssignals, Erlöschen des Leuchtmelders der das Schließens des Eingangssignals überwacht |

**Tableau 4.3** : États de fonctionnels de réalisation d'un itinéraire

**Tabelle 4.3** : Funktionelle Zustände bei dem Erstellen einer Fahrstraße

L'enclenchement de «Transit» assure l'enclenchement de aiguilles à la formation de l'itinéraire, maintient l'enclenchement des aiguilles lorsque l'itinéraire n'est plus formé jusqu'au dégagement de toutes les zones situées en amont et maintient le cas échéant l'enclenchement entre itinéraires de sens contraires. La libération d'un «Transit» est tributaire de la destruction de l'itinéraire, de la libération de toutes les zones en amont de la zone de transit et de la libération de sa zone propre.

Der so genannte „Transitverschluss" sichert den Weichenverschluss bei der Fahrstraßenbildung und hält, wenn die Fahrstraße nicht mehr gestellt ist, den Weichenverschluss aufrecht bis zum Freimachen aller davor befindlichen Bereiche; gegebenenfalls hält er den Verschluss zwischen Fahrstraßen entgegengesetzter Richtungen aufrecht. Die Freimachung eines „Transits" hängt von der Fahrstraßenauflösung, von der Freimachung aller Bereiche vor der Transitzone und von der Freimachung des eigentlichen Bereichs ab.

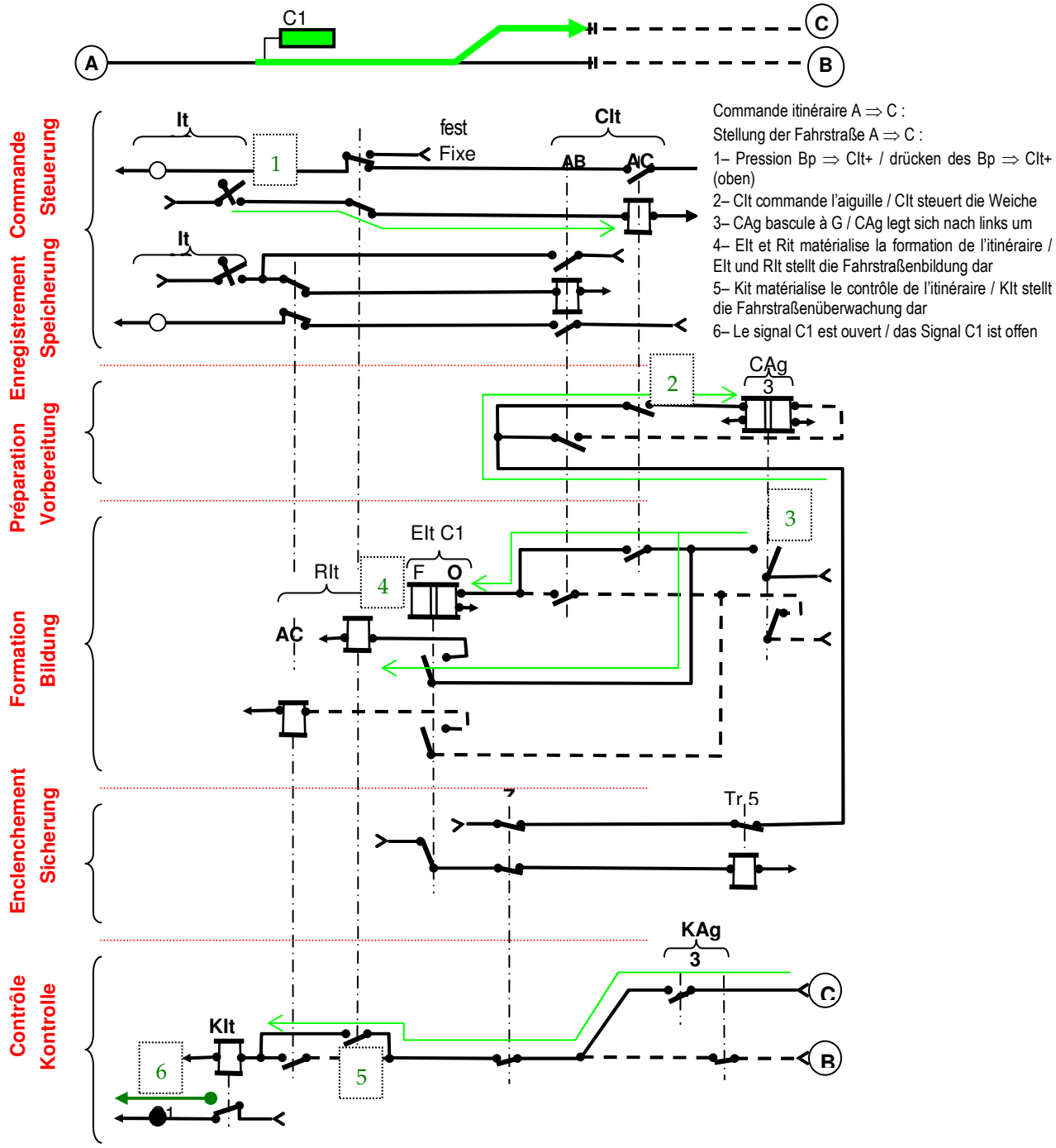



**Figure n°4.13** : Architecture générale des postes à itinéraire

**Abbildung n°4.13**: Allgemeine Architektur eines Fahrstraßenstellwerks

La réalisation d'un itinéraire est la conclusion de la réalisation et de l'achèvement de phases successives: (1) la commande de l'itinéraire, (2) son enregistrement, (3) sa préparation en commandant les ressources nécessaires, (4) sa formation en immobilisant les ressources positionnées au moyen de l'enclenchement de Transit, (5) le contrôle des ressources terrains de l'itinéraire, (6) la commande à l'ouverture du signal de protection.

Les fonctions de signalisation sont activées en couches successives qui prennent en compte les positions successives des organes de commande des ressources du poste, du positionnement et de l'occupation de ces ressources par les circulations.

## 4.5.3 L'architecture générale d'un poste d'aiguillage informatique

Depuis 1995 la SNCF a développé et mis en exploitation des postes informatiques dans le même cadre réglementaire. Ces postes reprennent donc les mêmes phases de fonctionnement que celles précédemment décrites [Gernigon, 1998] [Rétiveau, 1987].

L'architecture générale d'un poste d'aiguillage est donc le fruit de l'évolution historique (cf. Annexe B et figure 4.14) et peut décomposer en niveaux distincts assurant des fonctions de niveau de sécurité distincts.

Die Erstellung einer Fahrstraße gliedert sich auf in die Durchführung und den Abschluss verschiedener, aufeinander folgender Phasen: (1) Fahrstraßenanforderung, (2) Fahrstraßenspeicherung, (3) Fahrstraßenvorbereitung und Bereitstellung der notwendigen Ressourcen, (4) Fahrstraßenbildung und Immobilisierung der benutzten Ressourcen mittels einer Transitsicherung, (5) Überprüfung der Ressourcen der Fahrstraße vor Ort, (6) Öffnen des Deckungssignals. Die Signalfunktionen werden in aufeinanderfolgenden Ebenen aktiviert. Dabei werden die jeweilige Stellung der Steuerorgane der Ressourcen des Stellwerks und die Benutzung und Freigabe der Ressourcen der Strecke durch die Zugfahrt aktiviert.

## 4.5.3 Allgemeine Architektur eines rechnerbasierten Stellwerks

Seit 1995 entwickelt und betreibt die SNCF rechnerbasierte Stellwerke mit denselben Vorschriften. Diese Stellwerke übernehmen also exakt die oben beschriebenen Funktionsphasen [Gernigon, 1998] [Rétiveau, 1987].

Die allgemeine Architektur eines Stellwerkes ergibt sich also aus der historischen Entwicklung (siehe Anhang B und Abb. 4.14) und lässt sich in verschiedene Ebenen unterteilen, die Funktionen unterschiedlicher Sicherheitslevel ausführen.

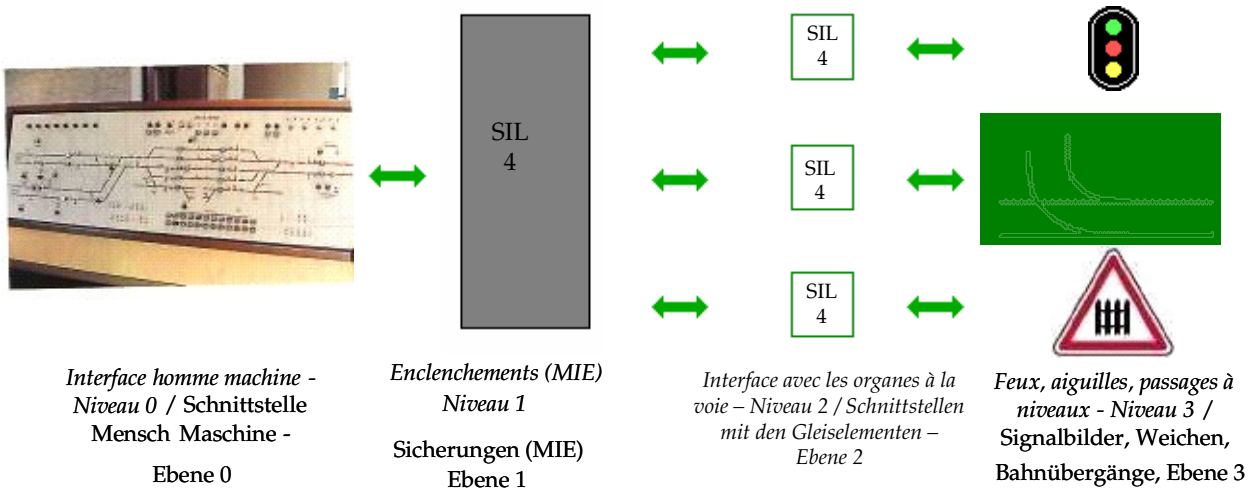



**Figure 4.14**: Architecture générale d'un poste d'aiguillage à itinéraire
**Abbildung 4.14**: Allgemeine Architektur eines Fahrstraßenstellwerks

**Niveau 0 : Aide à l'exploitation :** Interface homme machine, les fonctions correspondantes sont l'élaboration des commandes des itinéraires ou de certains organes particuliers et l'élaboration des contrôles visuels et sonores présentés à l'exploitant. Ce niveau N0 est réalisé avec une bonne sûreté de fonctionnement: niveau **SIL2**.

**Niveau 1 Enclenchements :** Son rôle consiste à assurer que les différentes commandes s'exécutent dans le respect des règles de sécurité. Ce niveau agit à la fois comme filtre sécuritaire des commandes issues du niveau 0 et générateur de dispositions complémentaires aux commandes reçues, dans le but d'assurer la sécurité. C'est le cœur de la sécurité du système. Ce niveau doit être réalisé avec le plus haut niveau de sécurité, **SIL4**.

**Niveau 2 Interface de commande et de contrôle des organes de la voie :** Moteurs d'aiguilles, signaux, circuits de voie. Les modes de défaillances doivent entraîner l'atteinte d'un état restrictif sûr (arrêt des circulations). Ce niveau doit être réalisé avec le plus haut niveau de sécurité, **SIL4**.

**Niveau 3 Organes de la voie :** Aiguilles, feux, Passages à niveaux. La baisse des coûts de l'électronique a entraîné une généralisation de l'informatique aux niveaux 0, 1 et 2. Les problèmes liés à cette informatisation sont essentiellement de deux types:

- la maîtrise des coûts complets d'installation impose de traiter :
  - les coûts des supports de transmission de données entre les différents niveaux ;
  - les coûts d'études d'un poste, chaque gare ayant une configuration matérielle et logicielle différente des autres gares, et chaque réseau ferroviaire ayant ses propres règles d'enclenchements ;
  - les faibles durées de vie des différents constituants, notamment lorsque les conditions d'environnement ne sont pas excellentes (climatisation, protection CEM…).

**Ebene 0**: **Unterstützung des Betriebs:** Schnittstelle Mensch-Maschine; die entsprechenden Funktionen sind die Fahrstraßensteuerung bzw. die Steuerung bestimmter Komponenten, sowie visuelle und akustische Kontrollen, die für das Betriebspersonal bestimmt sind. Diese N0-Ebene muss also eine gute Betriebssicherheit haben: **SIL2**.

**Ebene 1**: **Sicherung:** sie überprüfen, ob die unterschiedlichen Anforderungen auch die Sicherheitsregeln einhalten. Diese Ebene ist sowohl ein Sicherheitsfilter der Anforderungen aus der Ebene 0 als auch ein Erzeuger von die erhaltenen Befehle ergänzenden Bestimmungen, die die Sicherheit gewährleisten sollen. Sie ist der Kern der Systemsicherheit. Diese Ebene muss also mit dem höchsten Sicherheitslevel realisiert werden: **SIL4**.

**Ebene 2**: **Schnittstelle für die Steuerung und die Überwachung der Feldelemente:** Stellmotoren, Signale, Gleisstromkreise. Alle Fehler müssen zu einem sicheren, restriktiven Zustand führen (Fahrtunterbrechung). Diese Ebene muss also mit dem höchsten Sicherheitslevel realisiert werden: **SIL4**.

**Ebene 3: Feldelemente:** Weichen, Signalbilder, Bahnübergänge. Der Rückgang der Kosten für Elektronik hat eine Verallgemeinerung der Informatik auf Ebene 0, 1 und 2 bewirkt. Die Probleme, die mit dieser Informatisierung zusammenhängen, liegen hauptsächlich bei:

- der Beherrschung der vollständigen Einrichtungskosten; man muss dabei berücksichtigen:
  - die Kosten der Hardware für die Datenübermittlung zwischen den verschiedenen Niveaus
  - die Kosten von Stellwerksstudien; jeder Bahnhof hat eine andere Hardware-Softwarekonfiguration und jedes Eisenbahnnetz hat seine eigenen Sicherungsregeln
  - die geringe Lebensdauer der verschiedenen Bestandteile insbesondere dann wenn die Umweltbedingungen nicht hervorragend sind (Klimaanlage, elektromagnetischer Schutz…).

- la sécurité du système. Cette garantie se heurte à trois difficultés:
  - difficulté d'assurer que les programmes informatiques ne comportent pas d'erreur ;
  - difficulté de garantir l'élimination des erreurs d'origine matérielle lors de l'exécution des programmes ;
  - difficultés de réalisation d'une transmission de sécurité entre postes ou unités informatiques de sécurité. La sécurité du système repose sur le codage des informations échangées entre calculateurs de sécurité et sur le principe d'échange des informations (dynamisation et trivalence).

Parmi, les difficultés essentielles de conception des postes informatiques sont :

- l'identification de toutes les solutions dégradées possibles (elles sont très nombreuses et peuvent concerner à la fois la campagne et le poste) ;
- le besoin de sauvegarder de manière sûre en permanence la mémoire des états des installations pour permettre la réinitialisation du système sans créer de situation dangereuse ;
- la preuve d'un fonctionnement conforme aux objectifs de sécurité.

Les postes informatiques comportent des caractéristiques communes :

- le calcul des enclenchements par les unités de traitement travaillant en logique 2 parmi 3 ou en logique à 2 parmi 2 unités ;
- la mise en place systématique d'un système de gestion des protections (pour réalisation des travaux sur les voies).

Un module de paramétrage permet de saisir la situation du terrain pour la réalisation du poste et les modifications, Un module de simulation permet d'automatiser les tests de non régression en cas de modification des logiciels.

- der Sicherheit des Systems. Man stößt hier auf drei Schwierigkeiten:
  - die Schwierigkeit zu gewährleisten, dass die Programme keinen Fehler haben
  - die Schwierigkeit, die Eliminierung der Hardwarefehler bei der Ausführung der Programme zu garantieren
  - die Schwierigkeiten eine Sicherheitsübertragung zwischen Stellwerken oder IT-Sicherheitseinrichtungen. Die Sicherheit des Systems liegt im codierten Informationsaustausch zwischen den Sicherheitsrechnern und im Datenaustauschprinzip (Dynamische Aktualisierung und dreiwertige Logik).

Die wesentlichen Schwierigkeiten bei der Konzeption von Rechnerstellwerken sind:

- die Identifikation aller möglichen Rückfalllösungen (sie sind zahlreich und können sowohl die Anlage vor Ort als auch das Stellwerk betreffen)
- die Notwendigkeit, die sicheren Zustände der Einrichtungen permanent zu speichern, um das automatische Initialisieren des Systems zu erlauben, ohne einen gefährlichen Zustand hervorzurufen
- der Nachweis des Funktionierens gemäß den Sicherheitszielsetzungen.

Alle IT-Stellwerke haben gemeinsame Eigenschaften:

- Die Sicherheitsberechungen werden durch Bearbeitungseinheiten durchgeführt, die mit der 2 aus 3 Logik oder mit der 2 aus 2 Logik arbeiten.
- der systematische Rückgriff auf ein System zum Schutzmanagement (Durchführung von Gleisbauarbeiten).

Der notwendige Rückgriff auf eine Parametrierungseinheit erlaubt es, die Situation vor Ort bei der Realisierung des Stellwerks und gegebenenfalls bei Änderungen einzugeben. Eine Simulationseinheit erlaubt es automatisch zu testen, ob die Änderung der Software kein Rückschritt des Systems mit sich bringt.

Il est admis de nos jours qu'à condition que les frais de développement soient maîtrisés du fait d'une grande série, des économies peuvent être faites sur la réalisation. Toutefois, leur durée de vie est incertaine, notamment quand le poste doit être modifié après quelques années de mise en exploitation, ce alors que celle des postes à relais réalisés jusque dans les années 1980 dépasse 50 ans.

**Il est à noter que les principes des fonctions d'enclenchement des postes à relais à itinéraires ont été conservés mais réalisés différemment du fait du changement de technologie. De même, les noms et le vocabulaire ont été reconduits pour faciliter l'appropriation par les agents d'étude et de maintenance.**

## 4.5.4 Procédure de mise en service d'un poste d'aiguillage

Dans le processus de mise en exploitation d'un poste d'aiguillage, les tâches les plus chronophages pour les postes d'aiguillage informatiques sont celles de réalisation des essais avant mise en service. Pour être efficace sur les plans de l'économie et de la maîtrise du niveau de couverture, il faut donc agir essentiellement sur cette tâche.

Il est important de noter que le niveau de sécurité du système ferroviaire reposait, au moins en partie, sur la **validation formelle exhaustive** des postes mécaniques, électromécaniques et électriques reposant sur des opérateurs logiques dits de sécurité ou **fail safe** [Pichon, 1886] [Descubes, 1898]. Malheureusement ce n'est plus le cas des systèmes informatiques critiques, et c'est l'enjeu majeur des gestionnaires d'infrastructure pour les prochaines années, tant pour les postes informatiques que pour les systèmes tels qu'ERTMS… [Senesi, 2008]

**Notons que les plans de tests sont conçus pour vérifier que les fonctions de signalisation requises par le plan de voie sont présentes et correctement réalisées et non pas que les fonctions de signalisation présentes dans le poste d'aiguillage fonctionnent correctement ensemble.**

Es wird heute angenommen, dass, unter der Bedingung, dass die Entwicklungskosten aufgrund einer Massenproduktion begrenzt sind, Einsparungen bei der Realisierung gemacht werden können. Allerdings ist die Lebensdauer nicht genau bekannt, insbesondere wenn das Stellwerk einige Jahre nach Inbetriebnahme geändert werden muss. Die Lebensdauer von Relaisstellwerken hingegen, die bis in die achtziger Jahre gebaut wurden, beträgt mindestens 50 Jahre.

**Man stellt fest, dass die Grundsätze der Sicherungsfunktionen der Fahrstraßenrelaisstellwerke beibehalten, aber aufgrund des Technologiewechsels, anders umgesetzt wurden. Ebenso wurden Namen und technischen Begriffe beibehalten, um deren Aneignung durch das Test- und Wartungspersonal zu vereinfachen.**

## 4.5.4 Vorgehen bei Inbetriebnahme eines Stellwerks

Bei der Inbetriebnahme eines IT-Stellwerkes kostet die Durchführung der Versuche vor der Inbetriebnahme am meisten Zeit. Vom wirtschaftlichen Standpunkt und auch vom Abdeckungsgrad aus, muss man sich deshalb hauptsächlich mit dieser Aufgabe befassen.

Es ist wichtig anzumerken, dass das Sicherheitsniveau des Eisenbahnsystems zumindest teilweise auf der **kompletten, formalen Validierung** der mechanischen, elektromechanischen und elektrischen Stellwerke beruhte; diese wiederum beruhen auf logischen Sicherheits- bzw. **fail safe** Operatoren [Pichon, 1886] [Descubes, 1898]. Leider ist dies bei kritischen IT-Systemen nicht mehr der Fall, und das ist die wesentliche Herausforderung für die Infrastrukturbetreiber in den kommenden Jahren, sowohl für die IT-Stellwerke also auch für Systeme wie beispielsweise das ERTMS, usw. [Senesi, 2008]

**Die Testpläne werden für die Prüfung der Existenz und die Korrektheit des betreffenden Gleisplans konzipiert; dabei wird nicht geprüft, ob die im Stellwerk vorhandenen Signalgebungsfunktionen auch korrekt zusammen funktionieren.**

## 4.6 Exemples d'application

Insistons sur le fait qu'une installation (ou système) technique, conçu initialement, souvent implicitement, pour un cadre d'usage donné (condition, environnement, durée, fréquence…) peut se révéler «sure» ou «non sure» par rapport aux attentes d'une situation, d'un plan de voie et d'un programme fonctionnel donné. Illustrons ce constat par quelques exemples où l'installation technique peut être ou non «sure» selon la définition des postulats de fonctionnement retenus pour sa mise en œuvre.

### 4.6.1 Dispositif d'annonce des circulations aux chantiers

Le dispositif basé sur un détecteur d'annonce et un détecteur de libération sans continuité d'annonce se révèle, avec position sans annonce à la mise sous tension :

- sûr s'il est installé sur une ligne en block absolu (avec néanmoins l'acceptation du risque de non annonce en cas de pénétration en canton occupé) ;
- non sûr s'il est installé sur une ligne en block permissif (la seconde circulation n'est pas détectée et est encore plus dangereuse que la première de par son arrivée sur le chantier au moment où les agents entrent en voie) ;
- sûr si une procédure humaine requiert d'avoir l'assurance qu'aucune circulation ne se trouve entre les détecteurs à la mise sous tension du système, ce qui est réalisable en block non permissif ou en block manuel, ce qui est quasi impossible en block permissif automatique ;
- non sûr si aucune procédure de la sorte existe et est réellement applicable, ce qui est le cas en block permissif.

Il est à noter que les aspects «sûrs» et «non sûrs» sont **déterministes**, fonctionnels et non rien de probabiliste : dès que la configuration se présentera, le dispositif conduira de manière sure (avec son niveau de sécurité) à l'atteinte de cet état « sûr » ou « non sur » (la configuration correspond à l'arrivée d'un train sur le chantier).

Ceci explique pourquoi un tel système, pourtant mis en application depuis des années dans certains pays, n'est pas exploitable en l'état d'en d'autres : **la conception d'un système s'est effectué en cohérence homme produit procédure du pays. On ne peut pas sans précautions en extraire un bout et l'implanter dans un autre pays.**

## 4.6 Anwendungsbeispiele

Es muss betont werden, dass eine technische Anlage (bzw. System), die ursprünglich oft implizit für eine ganz bestimmte Verwendung (Bedingungen, Umfeld, Dauer, Frequenz…) konzipiert wurde, sich im Hinblick auf verschiedene Situationen, Gleispläne, funktionelle Programme als „sicher" oder als „unsicher" erweisen kann. Diese Feststellung wird im weiteren anhand von einigen Beispielen erläutert, bei denen die technische Anlage „sicher" oder „unsicher" sein kann, je nach Definition der für die Implementierung gewählten Anforderungen.

### 4.6.1 Zugwarnanlage für Baustellen

Die Vorrichtung, die auf einem Zugvormelder und auf einem Melder des Freiseins ohne Anzeigenkontinuität basiert, erweist sich, wenn sie sich bei Betriebaufnahme im Betriebszustand „ohne Anzeige" befindet, als:

- sicher, wenn sie auf einer Strecke mit absolutem Block installiert wird (gleichwohl mit der Akzeptanz des Risikos einer fehlenden Anzeige bei Einfahrt in einen besetzten Block).
- unsicher, wenn die Vorrichtung auf einem permissiven Block installiert ist (der nachfolgende Zug wird nicht gemeldet und ist umso gefährlicher, da er gerade dann ankommt, wenn das Personal die Baustelle betritt).
- sicher, wenn eine Vorschrift vom Personal fordert, zu überprüfen, dass sich bei der Betriebaufnahme des Systems kein Zug zwischen den Detektoren befindet, was in einem nicht permissiven oder manuellen Block realisierbar, in einem permissiven automatischen Block jedoch fast unmöglich ist
- unsicher, wenn kein derartiges Verfahren existiert und wirklich anwendbar ist, was beim permissiven Block der Fall ist.

Man stellt fest, dass die Eigenschaften „sicher" und „unsicher" **deterministisch**, funktionell und keinesfalls stochastisch sind: sobald die Konfiguration auftritt, wird die Vorrichtung zwangsläufig (auf ihrem Sicherheitsniveau) den „sicheren" oder „unsicheren" Zustand erreichen (die Konfiguration entspricht der Ankunft des Zuges auf der Baustelle). Dies erklärt, weswegen ein solches System, obwohl es in einigen Ländern seit Jahren in die Praxis umgesetzt wurde, in anderen Ländern nicht unverändert eingesetzt werden kann. **Die Entwicklung eines Systems wird in Übereinstimmung mit Mensch, Einrichtung und Vorschriften eines Landes durchgeführt. Man kann nicht einfach ohne Vorsichtsmaßnahmen einen Teil davon herausnehmen und diesen in einem anderen Land einsetzen.**

**L'analyse du risque requiert obligatoirement une triple culture : technique et réglementaire des pays (exemple figure 4.15).**

**Die Risikoanalyse erfordert zwingend eine dreifache Kultur: die der Technik und die der Vorschriften der beiden Länder (Beispiel Abb. 4.15).**

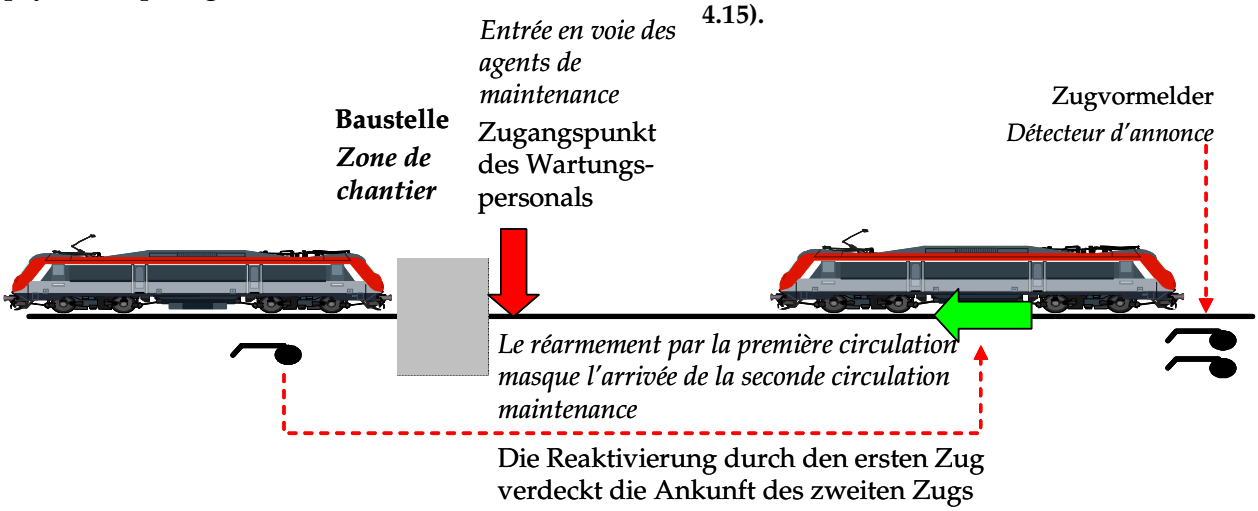



**Figure 4.15** : Conséquence d'une non prise en compte d'un postulat de fonctionnement (block permissif)

**Abbildung 4.15:** Folge einer Nichtbeachtung einer Funktionsanforderung (permissiven Block)

## 4.6.2 Les dispositifs de comptage d'essieux utilisés pour les installations de block automatique

Le dispositif basé sur le comptage séquentiel des essieux entrés et sortis d'une portion de voie, sans contrôle de la continuité d'annonce se révèle :

- sûr s'il est installé sur une ligne en block absolu (avec néanmoins l'acceptation du risque de libération à tord en cas de pénétration en canton occupé avec un disfonctionnement aléatoire du point de comptage d'entrée) ;
- non sûr s'il est installé sur une ligne en block permissif, la seconde circulation peut alors ne pas être détectée si le point de comptage d'entrée est en défaut sûr (occulté) lors du passage de la première circulation et que les deux circulations ont le même nombre d'essieux (Figure 4.16).

## 4.6.2 Achszähler für die automatische Blocksicherung

Die Vorrichtung, die auf der Zählung der ein- und ausfahrenden Achsen basiert und bei der die Anzeigenkontinuität nicht kontrolliert wird ist:

- sicher, wenn sie auf einer Strecke mit absolutem Block installiert wird (jedoch mit der Akzeptanz des Risikos einer irrtümlichen Freigabe im Falle der Einfahrt in einen besetzten Block und einem zufälligen Fehler des Eingangszählpunktes)
- unsicher, wenn sie auf einer Stecke mit permissivem Block installiert wird (der zweite Zug kann nicht gemeldet werden, wenn der Eingangszähler beim Überfahren des ersten Zuges im sicheren (nicht detektierten) Fehlerzustand ist und wenn die zwei Züge dieselbe Anzahl an Achsen haben. (Abb. 4.16)

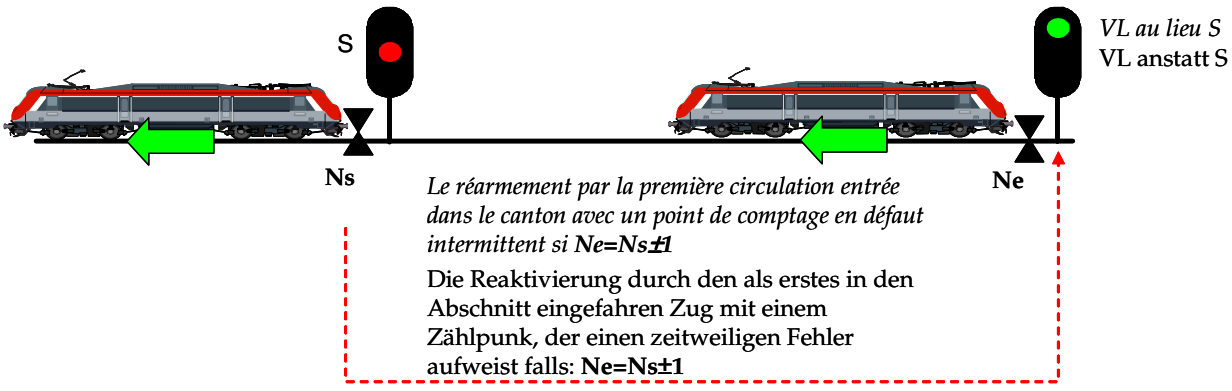



**Figure 4.16** : Conséquence d'une non prise en compte d'un postulat de fonctionnement
**Abbildung 4.16**: Folge einer Nichtbeachtung einer Funktionsanforderung

## 4.6.3 Les dispositifs de comptage d'essieux pour les zones de gare

L'usage de compteurs d'essieux en zone de gare afin de remplir des fonctions de transit souple a toujours été interdit en France. **Le fonctionnement de tels dispositifs repose sur le postulat suivant : «un train ne peut échapper à une portion de voie sans influencer un des points de comptage l'encadrant** ». Ce postulat n'est pas aisé à garantir en sécurité et sur la durée de vie du système[28]. La dépose des capteurs au rail (suite à des travaux de voie par exemple, Figure 4.17) n'est pas nécessairement détecté par le dispositif et conduirait à une entrée non détectée d'une circulation dans une portion de voie considérée libre. Une telle problématique serait encore aggravée par l'usage de transit souple : une circulation qui serait entièrement sur une zone considérée libre ne serait plus protégée d'aucune part !

## 4.6.3 Achszähler für Bahnhofszonen

Die Benutzung von Achszählern im Bahnhofsbereich bei Fahrstraßen mit Teilauflösung ist in Frankreich seit jeher verboten. **Die Funktionsweise solcher Anlagen beruht auf folgender Anforderung: „der Zug kann den Streckenabschnitt nicht verlassen, ohne einen der ihn umgebenden Zählpunkte zu beeinflussen"**. Es ist nicht leicht, diese Anforderung über die Lebensdauer des Systems hinweg sicher einzuhalten[29]. Die Anlage erkennt den Ausbau von Achszählern (beispielsweise nach Gleisbauarbeiten – Abb. 4.17) nicht unbedingt, und das kann dazu führen, dass ein Zug ungemeldet auf einen für frei geltenden Streckenabschnitt einfährt. Eine solche Problematik ist im Falle einer Teilfahrstraßenauflösung noch schlimmer: ein Zug, der sich vollständig auf einem als frei geltenden Streckenabschnitt befindet, ist von keiner Seite mehr geschützt!

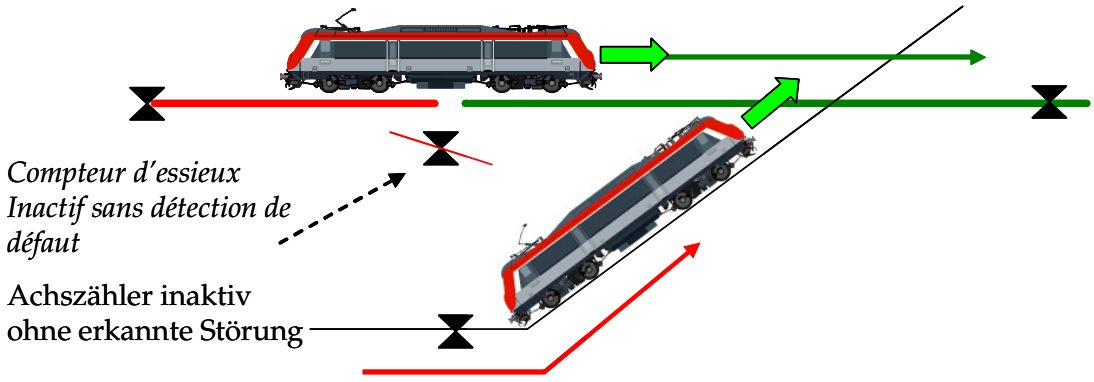



**Figure 4.17** : Situation dangereuse

**Abbildung 4.17**: Gefährliche Situation

## 4.6.4 Circulations ferroviaires et chocs aux passages à niveau

Le premier passage à niveau entièrement automatique est essayé en 1940 et la première campagne d'automatisation débuta en 1952. Il a été cherché des solutions techniques pour garantir que les barrières des passages à niveau se ferment à l'arrivée d'un train. Mais qu'on ne lui demande pas en plus de garantir que les automobilistes allaient respecter ces barrières ! Il y avait, dans sa conception de la sécurité, une hypothèse sous-jacente : **celle de la rationalité des tiers**. Nul n'est censé provoquer un accident de train, ni intentionnellement, ni par imprudence.

## 4.6.4 Eisenbahnverkehr und Zusammenstöße an Bahnübergängen

Der erste völlig automatische Bahnübergang wurde im Jahre 1940 getestet, und die erste Automatisierungskampagne begann im Jahre 1952. Es wurden technische Lösungen gesucht, die garantieren, dass die Schranken der Bahnübergänge sich beim Nähern eines Zuges schließen. Aber man hat sich nicht gefragt, ob die Autofahrer diese Schranken respektieren werden. Bei der Sicherheitskonzeption hat man also mit einer grundlegenden Hypothese gearbeitet: **die außenstehenden Personen sind vernünftig**. Niemand verursacht absichtlich oder unabsichtlich einen Zugunfall.

[28] En France : un point de comptage est obligatoirement constitué de deux capteurs électroniques implantés chacun sur une file de rail. L'actionnement isolé d'un des capteurs provoque l'engagement de l'intervalle (avec un nombre d'essieux inconnu le cas échéant). Ainsi, nombre de points de comptage étrangers ne sont pas compatibles avec ces règles de sécurité.

[29] In Frankreich: ein Achszähler besteht notwendigerweise aus zwei elektronischen Sensoren, die auf den beiden Schienensträngen installiert sind. Das Betätigen eines Sensors besetzt den Abschnitt (gegebenenfalls mit einer unbekannten Anzahl von Achsen). Die meisten ausländischen Achszähler sind nicht mit diesen Sicherheitsregeln kompatibel.

Techniquement, les passages à niveau automatiques en France (SAL) garantissent la fermeture des signaux routiers et des barrières au plus tard 10 secondes avant le passage de la circulation annoncée [Rétiveau, 1987]. Ceci :

- nécessite que les circulations ferroviaires soient étudiées pour qu'une collision sur un passage à niveau avec une circulation routière n'entraîne ni déraillement du premier bogie, ni n'occasionne de chocs tel qu'un passager puisse être blessé ;
- permet un usage sur les lignes en block permissif où les circulations peuvent circuler à distance d'arrêt l'une de l'autre.

En Allemagne, les passages à niveau sont conçus de manière à vérifier la fermeture des barrières avant d'autoriser l'arrivée de la circulation sur la chaussée (la probabilité de collision est donc plus faible qu'en France). Les contraintes appliquées à ces circulations sont donc plus légères, les rendant non compatibles avec une circulation en France. Il est à noter que de tels principes entraîneraient logiquement :

- un allongement des délais d'annonce afin de pouvoir arrêter la circulation ferroviaire en cas de non contrôle de fermeture des barrières. Ceci ne serait pas applicable sur des lignes en block automatique permissif ;
- un allongement des barrières interdisant leur franchissement sur toute la chaussée.

Ainsi, une même notion d'automatisme, ici de passage à niveau, peut recouvrir des réalités très diverses d'un pays à l'autre.

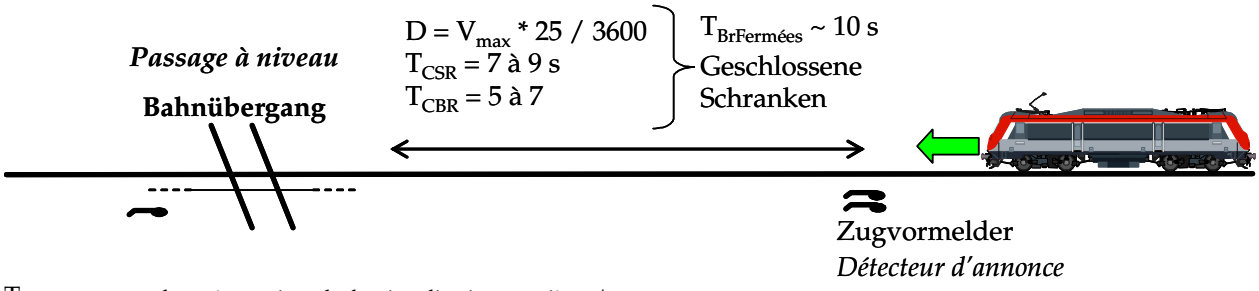
Technisch gesehen garantieren die automatischen Bahnübergänge in Frankreich (SAL) das Schließen der Straßensignale und der Schranken mindestens 10 Sekunden vor dem Vorbeifahren des angekündigten Zuges [Rétiveau, 1987]. Dies:

- erfordert, dass die Züge so konzipiert werden, dass ein Zusammenstoß auf einem Bahnübergang mit dem Straßenverkehr weder ein Entgleisen des ersten Drehgestells bewirkt, noch zur Verletzung eines Passagiers führen kann.
- erlaubt den Gebrauch der automatisierten Bahnübergänge auf permissiven Blocks, bei denen die Entfernung zum nächsten Zug nur der Bremsweg ist.

In Deutschland sind die Bahnübergänge so konzipiert, dass das Schließen der Schranken kontrolliert wird bevor der Zug die Fahrerlaubnis erhält (die Wahrscheinlichkeit eines Zusammenstoßes ist also niedriger als in Frankreich). Die für die Züge geltenden Regeln sind also weniger restriktiv und deshalb nicht mit dem Zugverkehr in Frankreich kompatibel. Solche Grundsätze würden logischerweise

- eine Verlängerung der Meldezeiten bewirken, um den Zugverkehr bei nicht durchgeführter Kontrolle der Schrankenschließung anhalten zu können. Dies wäre auf Strecken mit permissivem automatischem Block nicht machbar.
- eine Verlängerung der Schranken bewirken, damit das Überfahren des Bahnübergangs auf der ganzen Fahrbahnbreite verhindert wird.

So, kann dasselbe Automatismuskonzept, in diesem Fall der Bahnübergang, in verschiedenen Ländern unterschiedlich verwirklicht sein.




$T_{CSR}$ = temps de présentation de la signalisation routière / Dauer der roten Ampel

$T_{CBR}$ = temps de descente des barrières / Dauer der Schrankenschließung

**Figure 4.18** : Annonce au PN à SAL2 en France
**Abbildung 4.18:** Meldung bei einem SAL2-Bahnübergang in Frankreich

### 4.6.5 Fermeture automatique d'un signal de protection d'aiguille

Un signal de protection d'un itinéraire qui ne se refermerait pas lorsque la tête de la circulation ferroviaire le franchit, mais uniquement lorsque celle-ci atteint la fin de l'itinéraire protégé, peut être vu comme ayant un fonctionnement :

- **« sûr » si le block est non permissif** et interdit la présence d'une seconde circulation juste derrière la première. Dans ce cas, l'enclenchement des aiguilles peut être libéré au même moment (transit rigide) sans gêne pour les circulations ;
- **« non sure » si le block est permissif**. Il est alors nécessaire alors de limiter l'indication d'ouverture du signal à la seule première circulation, c'est le rôle de la fermeture automatique. Dans ce cas, l'enclenchement des aiguilles peut se libérer au fur et à mesure de l'avancement de la première circulation afin d'augmenter la souplesse du poste et le débit sur la première aiguille.

Il n'est pas possible d'évaluer les propriétés de sécurité d'une installation de sécurité sans connaître les postulats de fonctionnement tant règlementaire que physique.

### 4.6.6 Dispositif dit à usage contrôlé

Il est fait usage de dispositifs particulier dits à « usage contrôlé » pour permettre aux exploitants de gérer au mieux d'éventuels incidents de poste ou d'équipement terrain (par exemple circuit de voie, compteur d'essieux…). L'usage de ces dispositifs permet de s'affranchir d'un ou plusieurs enclenchements réalisés normalement dans le poste d'aiguillage. Leur usage n'est possible que dans le cadre d'une procédure règlementaire précise.

La sécurité du système complet, technique et opérateur nécessite la connaissance et la complémentarité des procédures règlementaires en vigueur pour exploiter le poste d'aiguillage.

### 4.6.5 Automatisches Schließen eines Weichendeckungssignals

Ein Fahrstraßendeckungssignal, das nicht schließt wenn die Zugspitze darüber fährt, sondern erst dann, wenn die Zugspitze das Ende der geschützten Fahrstraße erreicht hat, kann:

- **als „sicher" angesehen werden im Falle eines nicht permissiven Blocksystems** und falls ein zweiter Zug, direkt hinter dem ersten, verboten ist. In diesem Fall kann die Sicherung der Weichen zur gleichen Zeit freigemacht werden (Gesamtfahrstraßenauflösung), ohne Behinderung der Züge.
- **als „unsicher" angesehen werden im Falle eines permissiven Blocksystems**. Es ist notwendig, die Signalöffnung auf den ersten Zug zu beschränken. Dies ist die Aufgabe des automatischen Signalschlusses. In diesem Fall wird die Weichensicherung nach und nach durch die Fahrt des ersten Zuges freigegeben, um die Handhabung des Stellwerks zu flexibilisieren und den Durchsatz der ersten Weiche zu erhöhen.

Es ist nicht möglich, die Sicherheitseigenschaften einer Anlage zu bewerten, ohne die Anforderungen an die Funktionsweise (sowohl die vorschriftsmäßigen als auch die physischen) zu kennen.

### 4.6.6 Anlage in „überwachten" Betrieb

Es werden in Frankreich auch besondere Einrichtungen, so genannte Einrichtungen „unter überwachten Betrieb" genutzt; mit Hilfe derer behandelt das Betriebspersonal etwaige Störungen des Stellwerks oder einer Anlage vor Ort (z.B. in Verbindung mit einem Gleisstromkreis, einem Achszähler, usw.). Durch die Nutzung solcher Einrichtungen kann man auf einen bzw. mehrere Stellwerkverschlüsse verzichten. Dies ist jedoch nur im Rahmen einer präzisen Vorschrift möglich.

Die komplette Systemsicherheit (Technik und Bedienungspersonal) erfordert bei der Bedienung des Stellwerks sowohl die Kenntnis als auch die Komplementarität der Vorschriften.

## 4.7 Aspects à considérer en vue de réaliser une validation formelle

### 4.7.1 Possibilités d'indépendances et de quasi indépendances

Le fonctionnement d'un poste d'aiguillage à itinéraire, quelle que soit sa technologie, repose en France sur les mêmes phases de fonctionnement, à savoir :

- la commande : mémorisation de la volonté de l'opérateur de tracer un itinéraire ;
- la préparation : sollicitation des ressources à positionner pour assurer la réalisation et la protection de l'itinéraire mémorisé ;
- la formation : immobilisation (enclenchement) des ressources à réserver pour assurer la réalisation et la protection de l'itinéraire mémorisé ;
- le contrôle : vérification permanente des conditions d'établissement (contrôle des aiguilles, contrôle des protections, contrôle des particularités tels que les passages à niveau…).

Ainsi, sous réserve de vérifier que le processus d'étude n'altère pas ces indépendances entre itinéraires, nous pourrons utiliser ces indépendances ou quasi-indépendances pour réduire la combinatoire de l'exploration exhaustive. Il est à noter que ces propriétés sont actuellement utilisées pour la réalisation des essais des postes électromécaniques et électriques.

Dans ce qui suit, nous considérons comme quasi indépendant les informations ne participant au fonctionnel qu'en entrée d'une porte ET logique (fonction produit comme c'est le cas pour la fonction de contrôle KIt par exemple). Dans ce cas précis, il n'est pas nécessaire de tester l'ensemble des combinaisons des entrées. La figure 4.19 illustre ce point.

## 4.7 Zu beachtende Punkte bei der Durchführung einer formalen Prüfung

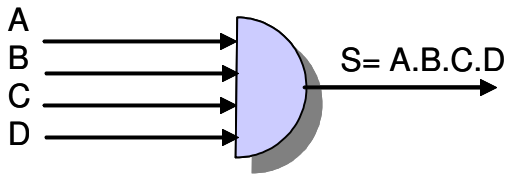
### 4.7.1 Möglichkeiten der Unabhängigkeiten und Fastunabhängigkeit

Das Funktionieren eines Fahrstraßenstellwerks beruht, ungeachtet seiner Technologie, auf denselben Funktionsphasen:

- Befehl: Speicherung der Entscheidung des Bedieners, eine Fahrstraße festzulegen
- Vorbereitung: Beanspruchung der benötigten Mittel, um die Verwirklichung und den Schutz der gespeicherten Fahrstraße zu gewährleisten
- Bildung: Sperrung (Sicherung) der notwendigen Mittel, um die Verwirklichung und den Schutz der gespeicherten Fahrstraße zu gewährleisten
- Kontrolle: permanente Prüfung der Realisierungsbedingungen (Kontrolle der Weichen, Kontrolle des Schutzes, Kontrolle der Besonderheiten wie z.B. Bahnübergänge…).

Somit können, unter der zu prüfenden Voraussetzung, dass sich diese Unabhängigkeiten zwischen den Fahrstraßen während der Signalisierung vor Prüfung nicht ändern, diese Unabhängigkeiten oder Fastunabhängigkeiten benützt werden, um die Kombinationsmöglichkeiten der vollständigen Untersuchung zu reduzieren. Es ist anzumerken, dass diese Eigenschaften heutzutage bei der Durchführung der Tests der elektrischen und elektromechanischen Stellwerke benützt werden.

Im Folgenden werden alle Informationen, die den Funktionen nur als Eingang einer UND-Logik dienen (UND-Funktion, wie zum Beispiel die Kontrollfunktion KIt) als fastunabhängig angesehen. In diesem Sonderfall ist es nicht notwendig, alle Eingangskombinationen zu testen. Abb. 4.19 illustriert dies.



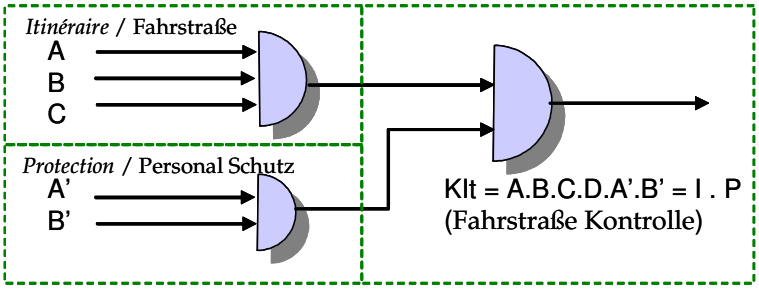

*Essais complets: $2^4$ combinaisons des entrées, 1 combinaison pour S=1 [1111] /*
Vollständige Tests: $2^4$ Eingangskombinationen, 1 Kombination für S=l [1111]

*Essais suffisants : N+1 combinaisons à tester /*
Ausreichende Tests: N+l zu testende Kombinationen
[1110] [1101] [1011] [0111] [1111]

**Figure 4.19** : Exemple de quasi-indépendance / **Abbildung 4.19**: Beispiel einer Fastunabhängigkeit

Cette démarche utilisée sur plusieurs niveaux (Figure 4.20).

Diese Vorgehensweise wird auf verschiedenen Ebenen genutzt (Abb. 4.20).




*Essais suffisants :*
*(N1+1) + (N2+1) + (N3+1)*
*avec N1=3, N2=2, N3=2 : N=10*
*[111] [110] [101] [011] Itinéraire*
*[11] [10] [01] Protection*
*[11] [10] [01] Kit*
*Essais complet : N* $= 2^5 = 32$

Ausreichende Versuche
*(N1+1) + (N2+1) + (N3+1)* Mit N1=3, $N_2$=2, $N_3$=2 : N=10
[111] [110] [101] [011] Fahrstraße
[11] [10] [01] Schutz
[11] [10] [01] KIt

**Figure n°4.20** : Exemple de quasi-indépendance avec deux niveaux
**Abbildung 4.20**: Beispiel einer Fastunabhangigkeit mit zwei Ebenen

Il vient de ce qui précède que l'on peut envisager de réaliser indépendamment les essais de chaque itinéraire ayant le même signal origine, de même.

**Nous verrons plus avant que cette possibilité de « découpage » du fonctionnel applicatif va permettre de réaliser plusieurs preuves complémentaires (et indépendantes dans leur réalisation).**

Il est important de noter que cette possibilité requiert la connaissance de la structure des graphes fonctionnels du poste d'aiguillage (cf. paragraphe avec les étapes de fonctionnement), leur vérification par une analyse des dépendances et des graphes du fonctionnel.

Cette connaissance n'est accessible qu'en conception manuelle avec le respect des principes d'établissement des fonctions de signalisation, ce qui exclut les langages en écriture automatique de code.

Par ailleurs, cette connaissance *a priori* est largement utilisée dans la réalisation manuelle des essais avant mise en service. La démarche générale est alors celle d'une validation formelle exhaustive des fonctions de l'architecture générale des fonctions du poste.

Daraus folgt, dass man erwägen kann, alle Fahrstraßen, die beim gleichen Ursprungssignal anfangen, unabhängig und auf dieselbe Weise zu testen.

**Es wird später gezeigt, dass es diese Aufgliederungsmöglichkeit der funktionellen Anwendungssoftware erlauben wird, mehrere ergänzende (und in ihrer Durchführung unabhängige) Tests durchzuführen.**

Es muss unterstrichen werden, dass diese Möglichkeit die Kenntnis der Struktur der funktionellen Schaubilder des Stellwerks (siehe Abschnitt 7.4.2 über die Funktionsetappen) sowie ihre Prüfung durch eine Abhängigkeitsanalyse und die Kenntnis der Graphen der Funktionen erfordert

Diese Kenntnis ist nur in manueller Entwicklung unter Beachtung der Grundsätze der Signalfunktionsbildung zugänglich, was Sprachen mit automatischer Codeerzeugung ausschließt.

Zusätzlich wird diese *a priori* Kenntnis viel bei der manuellen Durchführung der Versuche vor Inbetriebnahme benutzt. Das allgemeine Vorgehen ist dann eine vollständige formale Überprüfung der Funktionen basierend auf der allgemeinen Architektur der Stellwerksfunktionen.

## 4.7.2 Postulats de fonctionnement

Les exemples du paragraphe 4.6 nous ont montré qu'avant de procéder à une validation, formelle ou manuelle, il est impératif de formaliser toutes les conditions définissant le domaine de fonctionnement sûr. Ces conditions limitent le domaine des possibles des entrées du système :

- La réglementation limite les possibilités des opérateurs et des maintenances [Rétiveau, 1987] ;
- Les installations extérieures limitent elles-mêmes les possibilités d'évolution des entrées[30] ;
- Les modes de défaillances sures des équipements extérieures.

Quelque soit la méthode de validation d'une installation, il est nécessaire de définir le domaine possible des entrées du système. Nous verrons qu'il est indispensable afin de fixer les limites d'exploration automatique des états systèmes atteignables. Les postulats de fonctionnement sont de différentes natures. Citons notamment :

- le respect des procédures par les différents opérateurs intervenant ou utilisant le système à valider :
  - le mécanicien ne recule jamais sans ordre (cette action risquerait de libérer à tort les enclenchements de transit). C'est un choix, il serait aujourd'hui possible de lever cette contrainte dans les zones d'action des postes informatiques ;
  - les opérateurs exploitant le poste n'utilisent les dispositifs mis à leur disposition que dans le cadre des réglementations et des procédures, notamment les dispositifs lui permettant de s'affranchir de certains enclenchements en cas défaillance de prise d'information ;
  - les opérateurs ne délivrent des bulletins de franchissement qu'après vérification manuelle de toutes les conditions de sécurité précisées exhaustivement dans la consigne du poste (consigne rose du poste) ;
- la prise en compte des limitations des entrées du fait de l'existence de fonctions d'enclenchement câblées extérieurement au système à valider. Il s'agit par exemple des circuits de contrôle impératif d'application des lames d'aiguilles qui ne peuvent être établis simultanément à gauche et à droite, des circuits de report qui sont activés de manière sûre lors de l'extinction d'un feu commandé ;

[30] Par exemple, les circuits électriques interdisent la mise en position « vrai » des contrôles à gauche et à droite

## 4.7.2 Funktionelle Grundprinzipien

Die Beispiele des Abschnitts 4.6 haben gezeigt, dass es vor der formalen oder manuellen Überprüfung unbedingt notwendig ist, alle Bedingungen die den sicheren Funktionsbereich definieren, zu formalisieren. Diese Bedingungen beschränken die möglichen Systemeingänge:

- das Regelwerk beschränkt die Möglichkeiten des Bedienungs- und Instandhaltungspersonals [Rétiveau, 1987]
- die Außenanlagen selbst beschränken die Veränderungsmöglichkeiten der Eingänge[31]
- die sicheren Ausfallmodi der Außenanlagen beschränken ebenfalls die Systemeingänge.

Welches Verfahren auch immer zur Überprüfung der Anlage benützt wird, der Bereich der möglichen Systemeingänge muss bestimmt werden. Wie noch gezeigt werden wird, ist dies zur Beschränkung der automatischen Abtastung der erreichbaren Systemzustände unabdingbar. Die Anforderungen an die Funktionsweise sind sehr unterschiedlich, zum Beispiel:

- Einhaltung der Vorschriften durch das Bedienungspersonal des zu prüfenden Systems:
  - Der Fahrzeugführer fährt niemals ohne Befehl rückwärts (dadurch könnten Fahrstraßenverschlüsse fälschlicherweise aufgelöst werden). Dies ist eine Annahme die getroffen wurde, denn es wäre heute möglich, diese Beschränkung in den Aktionsbereichen der IT-Stellwerke aufzuheben.
  - Das Bedienungspersonal des Stellwerks nutzt die Notfallanlagen nur im Rahmen der Vorschriften und Regelwerke (insbesondere wenn diese Notfallanlagen bei einem Ausfall Verschlüsse aufheben können).
  - Das Bedienungspersonal erstellt erst dann den Befehl zum Überfahren eines Haltesignals, wenn es alle in der Hauptdienstanweisung des Stellwerks vollständig präzisierten Bedingungen manuell überprüft hat.
- Berücksichtigung der Eingangsbeschränkungen aufgrund bestehender Verschlussfunktionen, die extern mit dem geprüften System verkabelt sind. Dabei handelt es sich beispielsweise um Prüfungsstromkreise der Weichenzungen, die nicht gleichzeitig links und rechts wahr sein können, oder um Meldeschaltungen, die sicher aktiviert werden, falls ein angesteuertes Signal erlischt.

[31] Zum Beispiel, verbieten die Stromkreise die gleichzeitige Stellung des rechten und des linken Weichenprüfers auf die Position „wahr".

- les circulations se déplaçant sur le plan de voie activent chronologiquement les capteurs de présence (dispositifs de détection ponctuels ou continus) ;
- les entrées terrain ne participant qu'à une seule couche de fonctionnement (par exemple la couche de contrôle des itinéraires) n'ont pas besoin d'être activés quand la connaissance *a priori* de l'architecture des fonctions de signalisation montre que leur activation sera sans effet sur le fonctionnement du poste ;
- les dispositifs extérieurs assurant des fonctions dites de sécurité (SIL4) ont une position de repli sûre en cas de défaillance. Les comportements non sûrs de ces dispositifs n'ont donc pas â être considérés.

D'une manière générale, les postulats traduisent :

- les conditions organisationnelles, procédurales et/ou réglementaires de l'exploitation d'un poste d'aiguillage ;
- les conditions d'interactions des circulations ferroviaire avec les postes d'aiguillage (au niveau des ces capteurs et des ressources commandées notamment) ;
- les conditions d'interaction des installations extérieures, ressources commandées et/ou postes encadrants, avec le poste d'aiguillage.

Il apparaît clairement que ces postulats peuvent varier significativement d'un pays à l'autre, d'un type de circulation à un autre, d'un environnement à un autre. C'est la raison pour laquelle, la mise en œuvre d'un système réputé développé correctement pour apporter les assurances ad hoc pour un pays, n'est pas nécessairement acceptable dans un autre pays.

**De plus, ces postulats de fonctionnement à prendre doivent être définis de manière générique pour un itinéraire donné. Ces règles vont alors être utilisées pour délimiter l'espace des possibles des entrées qui limitera celui des états systèmes atteignable dans le fonctionnel du poste d'aiguillage.**

**Les postulats de fonctionnement reposent sur la formation des agents et sur le compagnonnage «métier». Ces postulats formalisent des connaissances métiers qui, dans une très large mesure, sont indépendants de la nature informatique des installations.**

**Il est à noter que ces postulats sont inconnus ou partiellement méconnus des industriels fabricant les postes d'aiguillage modernes.**

- Die Züge auf den Streckenabschnitten aktivieren nach und nach die Sensoren (zur punktuellen bzw. ständigen Zugortung).
- Die Eingänge der Außenanlagen, die nur eine Funktionsebene betreffen (z. B. die Fahrstraßenkontrollebene), brauchen nicht aktiviert zu werden, wenn die *a priori* Kenntnis der Signalsystemsfunktionsarchitektur zeigt, dass deren Aktivierung die Stellwerksfunktionsweise nicht beeinflusst.
- Außenanlagen, die Sicherheitsfunktionen (SIL4) ausführen, haben bei einem Ausfall eine sichere Rückfallebene. Unsichere Verhalten solcher Anlagen müssen also nicht berücksichtigt werden.

Im Allgemeinen umfassen die Anforderungen folgende Bedingungen:

- Bedingungen in Verbindung mit der Organisation, den Vorschriften und/oder den Regelwerken für den Betrieb des Stellwerks
- Bedingungen der Wechselwirkungen zwischen den Zügen und den Stellwerken - (insbesondere auf der Ebenen der Sensoren und der angesteuerten Ressourcen)
- Bedingungen der Wechselwirkungen zwischen den Außenanlagen, den angesteuerten Ressourcen oder den Nachbarstellwerken und dem Stellwerk.

Es ist einleuchtend, dass sich diese Anforderungen von einem Land zum anderen, von einer Zugart zur anderen und von einem Umfeld zum anderen unterscheiden. Deswegen ist ein für ein bestimmtes Land korrekt entwickeltes System, das in einem Land direkt zur Sicherheit beiträgt, in einem anderen Land nicht unbedingt akzeptierbar.

**Ferner müssen diese Anforderungen für eine bestimmte Fahrstraße allgemein definiert werden. Diese Regeln dienen der Begrenzung der möglichen Eingänge und somit der erreichbaren Systemzustände in den Stellwerksfunktionen.**

**Die Anforderungen an die Funktionsweise beruhen auf der Schulung der Bediensteten und auf dem Berufsstand. Diese Anforderungen formalisieren Fachkenntnisse, die weitgehend unabhängig von dem IT-System der Anlagen sind.**

**Ferner ist anzunehmen, dass diese Anforderungen in der Industrie (Hersteller von modernen Stellwerken) unbekannt sind bzw. teilweise die berufliche Kenntnis ihres „Metiers" verkannt wird.**

## 4.7.3 Propriétés de sécurité et incompatibilités

Comme nous l'avons vu dans les paragraphes précédents, les incompatibilités à réaliser pour assurer la sécurité de l'exploitation sont réalisées au moyen de deux « enclenchements » correspondants :

- les incompatibilités traduisent les exclusions d'états de ressources, physiques ou non, gérées par les installations de sécurité ;
- Les incompatibilités binaires sont réalisées aux moyens de deux enclenchements, les enclenchements conditionnels ternaires sont réalisés au moyen de neufs enclenchements.

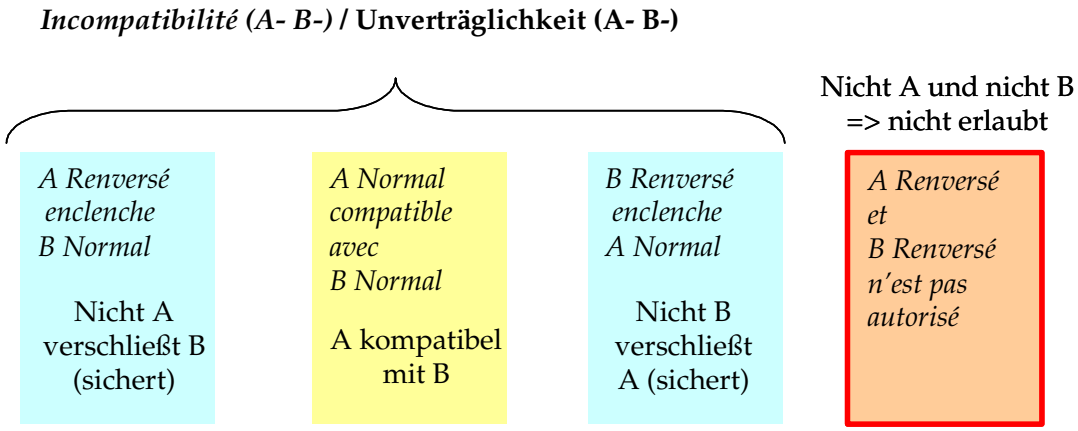
D'une manière générale, quelque soit la technologie utilisée, une incompatibilité est opérée par la réalisation de l'ensemble des enclenchements correspondants. La figure 4.21 illustre cela :

## 4.7.3 Sicherheitseigenschaften und Inkompatibilität

Wie bereits in den vorangegangenen Abschnitten gezeigt, werden die für die Gewährleistung der Betriebssicherheit notwendigen Inkompatibilitäten durch zwei entsprechende Sicherungen erzeugt:

- Die Unverträglichkeiten zeigen den Ausschluss von Ressourcenzuständen auf, die von den Sicherheitsanlagen gesteuert werden.
- Binäre Unverträglichkeiten werden durch zwei Sicherungen erstellt, die bedingten dreifachen Sicherungen bestehen aus neun Verschlüssen.

Ganz allgemein ist es so, dass bei jeder angewandten Technologie die Unverträglichkeit mit Hilfe der Durchführung aller entsprechenden Sicherungen erzeugt wird. Abb. 4.21 zeigt dies.




**Figue 4.21** : Réalisation d'une incompatibilité binaire

**Abbildung 4.21**: Binäre Unvertraglichkeit

Par reconduite des principes de signalisation précédemment présentés aux paragraphes précédents, les propriétés de sécurité se déduisent directement des fonctions d'incompatibilité et/ou d'enclenchement exigées par le plan de voie. Il s'agit notamment des fonctions identifiées plus avant :

- enclenchement de formations d'itinéraire ;
- enclenchement de contrôle du bon positionnement des ressources nécessaire à un itinéraire ;
- enclenchement d'approche ;
- enclenchement de transit ;
- enclenchement d'immobilisation des aiguille ou autres ressources ;
- enclenchement de sens, de voie unique ou d'affrontement…

Durch die Beibehaltung der zuvor präsentierten Signaltechnikgrundsätze ergeben sich die Sicherheitseigenschaften direkt aus den Unverträglichkeitsfunktionen und/oder aus den vom Gleisplan geforderten Verschlüssen. Es handelt sich insbesondere um die zuvor identifizierten Funktionen:

- Sicherung der Fahrstraßenbildung,
- Sicherung der Kontrolle der richtigen Lage der für eine Fahrstraße notwendigen Ressourcen,
- Annäherungsverschluss,
- Fahrstraßensicherung,
- Weichenblockierungssicherung oder Sicherung der Blockierung anderer Ressourcen,
- Richtungssicherung, Sicherung einer eingleisigen Strecke, Sicherung einer Frontalfahrt etc.

Leur traduction est nettement plus aisée que celle des postulats de fonctionnement, notamment du fait de la formalisation (et la conservation) rigoureuse des principes de signalisation à la SNCF depuis de nombreuses années.

Il est à noter que l'augmentation de la part de développement des systèmes confié aux industriels et le fort taux de renouvellement des agents d'étude conduit à la remise en cause de principes dans certaines technologies.

**Ainsi, alors que par le passé un principe consistait à définir les « principes d'enclenchement » indépendamment de la technologie choisie pour la réalisée, force est de constater que celui-ci disparaît peu à peu en pratique, conduisant à l'apparition de situations potentiellement dangereuse à terme.**

## 4.7.4 Propriétés de non surabondance

**Les incompatibilités assurant la sécurité ne doivent pas entraver les fonctionnalités attendues[32].** L'existence d'enclenchements parasites ou superflus peut être source d'insécurité :

- par application de procédures manuelles règlementaires par les opérateurs ;
- par le fait montre une non-conformité de l'installation de sécurité, celle-ci pouvant masquer des défauts non sûrs non encore apparus (accès à des états système non prévus).

Comme nous l'avons vu lors des chapitres précédents, la réduction de la disponibilité opérationnelle d'un poste d'aiguillage à un impact certain sur son niveau global de sécurité. Aussi, toute réduction indue au sens de la sécurité des possibilités du poste doit être détectée et traitée. Il s'agit d'offrir aux exploitants une installation la plus disponible et souple possible, afin de réduire l'occurrence des situations requérant l'application de procédures réglementaires.

Il est nécessaire de trouver les situations anormalement restrictives qui pourraient indirectement être cause d'insécurité, soit par le biais d'une application réglementaire, soit en masquant potentiellement d'autres défauts.

Notons qu'un signal restant toujours en position de fermeture respectera toujours les propriétés de non ouverture simultanée avec un autre signal, mais ceci ne garantit pas le fonctionnement sûr du poste.

[32] Propriétés surabondante: se dit de propriétés pouvant être retirées du fonctionnel sans que la sécurité du système soit réduite

Deren Umsetzung ist viel leichter als diejenige der Funktionsanforderungen, unter anderem aufgrund der langjährigen strengen formalen Gestaltung (und der Beibehaltung) der Grundsätze des SNCF-Signalsystems.

Es ist anzumerken, dass der Anstieg des Anteils der der Industrie anvertrauten Systementwicklungsarbeiten und der neu eingestellten Entwicklungsingenieure dazu führt, dass Prinzipien bestimmter Technologien in Frage gestellt werden.

**Das in der Vergangenheit geltende Prinzip, die „Sicherungsprinzipien" unabhängig von der Technologie zu definieren, wird nach und nach in der Praxis aufgegeben; dies führt längerfristig zu potentiell gefährlichen Situationen.**

## 4.7.4 Eigenschaften der Überflüssigkeit

**Die Sicherheit gewährleistenden Unverträglichkeiten dürfen die erwarteten Funktionen nicht beeinträchtigen[33].** Das Bestehen von störenden oder überflüssigen Sicherungen kann zu Unsicherheit führen:

- durch die Anwendung von manuellen Vorschriften durch das Bedienungspersonal
- durch die Nichtkonformität der Sicherheitsanlage, die noch nicht aufgetretene, unsichere Fehler verbergen können (Zugang zu unvorhergesehenen Systemzuständen).

Wie bereits erörtert, beeinflusst die reduzierte betriebliche Verfügbarkeit des Stellwerks unbestreitbar das Gesamtsicherheitsniveau. Jede ungerechtfertigte Verminderung der Stellwerkskapazität in Bezug auf die Sicherheit muss festgestellt und behandelt werden. Das Betriebspersonal muss eine Anlage haben, die so verfügbar und so flexibel wie möglich ist, damit die Anzahl von Situationen, die eine Anwendung der Vorschriften erforderlich machen, reduziert wird.

Es müssen anormale restriktive Situationen gefunden werden, die indirekt zur Unsicherheit führen könnten, entweder im Rahmen der Anwendung der Vorschriften oder durch das potentielle Verbergen anderer Fehler.

Es ist anzumerken, dass ein andauernd geschlossenes Signal zwar dazu führt, dass dieses Signal niemals gleichzeitig mit einem anderen Signal geöffnet wird, was jedoch nicht die sichere Funktionsweise des Stellwerks gewährleistet.

[33] Überflüssige Sicherheitseigenschaft : man kann auf diese Eigenschaft verzichten, ohne die Sicherheit des Systems zu reduzieren

## 4.8 Ce qu'il faut retenir pour la suite du travail

Notons des éléments clés qu'il faudra considérer ultérieurement pour réaliser la validation formelle d'un fonctionnel ferroviaire de sécurité :

- le système ferroviaire d'un pays est un système rendu cohérent par l'histoire et la pensée nationale, un système complexe qu'il n'est pas aisé de transposer en maintenant la sécurité des circulations ;
- il est illusoire de vouloir appliquer quelque méthode formelle que ce soit sans comprendre et formaliser un ensemble de règles et/ou de considérations physiques, explicites et largement implicites, techniques et systèmes, humaine et règlementaire. Ces règles sont largement méconnues des industriels modernes ;
- il est illusoire de vouloir appliquer une méthode formelle sur un système existant (intégré dans un contexte existant) sans en connaître les indépendances et/ou dépendances existants ainsi de fait ;
- toutes ces règles, dépendances, considérations physiques, considérations systèmes, considérations humaines… se traduisant sous formes de postulats, de propriétés fonctionnelles et propriétés de sécurité. Quelle que soit la méthode formelle utilisée ultérieurement cette explicitation et cette formalisation sont obligatoires et constituent la difficulté majeure de ce travail de validation formelle ;
- pour faciliter ce travail de formalisation, il est indispensable de faire participer des experts du « métier » et non des ingénieurs SdF : un travail collaboratif entre les deux métiers est indispensable. A cette fin, il conviendra lors du choix de la méthode formelle de rechercher la plus grande capacité de communication, la plus grande facilité d'appropriation ;
- il est à noter qu'un système donné (informatique ou non) peut être considéré «sûr» ou «non sûr» en fonction du contexte, des modes d'exploitation, des défaillances possibles ou non de certains capteurs… la sécurité n'est pas dans le système informatique mais dans le «système» ;
- il ne peut y avoir de preuve formelle d'un système indépendamment du contexte métier : Ainsi avant de faire de la conception validation de système informatiques pour le ferroviaire, il faut apprendre le ferroviaire dans toute sa généralité et complexité.

## 4.8 Weiteres Vorgehen

Im Folgenden werden einige Schlüsselelemente aufgeführt, die später bei der formalen Validierung der Bahnsicherheitsfunktionen zu beachten sind:

- Das Bahnsystem eines bestimmten Landes steht im Einklang mit seiner Geschichte und der dort vorherrschenden Denkweise; es handelt sich um ein komplexes System, das unter Beibehaltung der Zugssicherheit nicht leicht übertragen werden kann.
- Die Anwendung eines formalen Verfahrens ist illusorisch, wenn man nicht versucht, alle physischen, expliziten und weitgehend impliziten, menschlichen und vorschriftsmäßigen, technischen und systembezogenen Regeln und/oder Betrachtungsweisen zu verstehen und zu formalisieren. Diese Regeln sind den heutigen Herstellern weitgehend unbekannt.
- Die Anwendung eines formalen Verfahrens auf ein bestehendes System (in einem bestehenden Umfeld) ohne Kenntnis der Abhängigkeiten und/oder der Unabhängigkeiten, ist also de facto illusorisch.
- All diese Regeln, Abhängigkeiten, physische Betrachtungen, Systembetrachtungen, menschliche Betrachtungen, usw. können in Form von Anforderungen, Funktionseigenschaften oder Sicherheitseigenschaften ausgedrückt werden. Welches formale Verfahren später auch immer verwendet wird, die Eindeutigkeit und die Formalisierung sind unverzichtbar, sie bilden die Hauptschwierigkeit im Rahmen der formalen Überprüfung.
- Zur Erleichterung der Formalisierung ist es unabdingbar, Experten des Fachgebiets einzubeziehen, und nicht nur Experten der Funktionssicherheit: beide Berufsgruppen müssen zusammenarbeiten. Bei der Wahl des formalen Verfahrens muss man also auf die Verständlichkeit und leichte Aneignungsbarkeit der Methode achten.
- Ferner ist anzumerken, dass ein bestimmtes System (sei es ein IT-System oder nicht) in Abhängigkeit vom Umfeld, von der Betriebsart, vom möglichen Ausfall bestimmter Sensoren, usw. sowohl „sicher“ als auch „unsicher“ sein kann. Die Sicherheit liegt nicht im IT-System, sondern im „System“ selbst.
- Unabhängig vom Fachgebiet gibt es keine formale Überprüfung eines Systems: also bevor man IT-Systeme für die Eisenbahn konzipiert und validiert, muss man erst einmal die Eisenbahn ganz allgemein in all ihrer Komplexität kennenlernen

# CHAPITRE 5

# État de l'art des méthodes de validation des systèmes critiques – Introduction aux méthodes formelles

## 5.1 Sûreté de fonctionnement des systèmes programmés / état actuel

### 5.1.1 Besoins et enjeux

L'omniprésence des logiciels dans tous les systèmes industriels, et notamment des systèmes devant répondre à des exigences fortes en matière de sûreté de fonctionnement, implique que tout industriel se dote des moyens de « maîtriser » les logiciels qui le concernent et soit en mesure de démontrer cette maîtrise qui répond à l'obligation de moyens requis par les normes.

Pour essayer de pallier les risques inhérents aux logiciels de ces systèmes qui sont eux aussi porteurs de contraintes de sûreté de fonctionnement, les industriels du logiciel, les instances normatives et les grands constructeurs ont, depuis quelques années, établi des référentiels normatifs en matière de développement logiciels à fortes contraintes de sûreté de fonctionnement. Certains domaines (aéronautique, ferroviaire, nucléaire), se sont dotés d'instances chargées de vérifier la conformité aux référentiels normatifs du domaine et sensées quantifier le «niveau» de sûreté de fonctionnement obtenu par ces logiciels et par les systèmes dans lesquels ils sont intégrés.

L'objectif de cette première partie est de mettre en exergue les dispositions globales et les principes directeurs qui président aujourd'hui à tous ces référentiels métiers pour la production de logiciels à fortes exigences de sûreté de fonctionnement.

# KAPITEL 5

# Stand der Technik für die Überprüfungen kritischer IT-Systeme - Einführung in die formalen Methoden

## 5.1 Sicherheit programmierter Systeme / heutiger Stand

### 5.1.1 Bedürfnisse und Herausforderungen

Die Allgegenwärtigkeit von Software in allen Industriesystemen, die hohen Sicherheitsanforderungen nachkommen müssen, fordert von jedem Unternehmen sich mit Mitteln auszustatten, um die entsprechende Software „zu beherrschen". Das Unternehmen muss in der Lage sein, diese Beherrschung nachzuweisen (nachzuweisen, dass es den Anforderungen der Normen entspricht).
Die Softwareentwickler, die Normungsinstanzen und die großen Unternehmen haben seit einigen Jahren normative Bezugssysteme für die Entwicklung von Software mit hohen Funktionssicherheitsansprüchen aufgestellt, um die Risiken, die der Software dieser Systeme eigen sind, zu reduzieren. Bestimmte Gebiete (Luftfahrt, Eisenbahn, Atomenergie), haben sich mit Instanzen ausgestattet, die die Übereinstimmung des Fachgebietes mit den normativen Bezugssystemen überprüft. Diese Instanzen sollen das „Niveau" der Funktionssicherheit quantifizieren, das durch diese Software und durch die Systeme erreicht wird.

Die Zielsetzung dieses ersten Teils besteht darin, die globalen Bestimmungen und die leitenden Grundsätze hervorzuheben, die heute all diesen Bezugssystemen für die Entwicklung von Software mit hoher Funktionssicherheit zugrunde liegen.

## 5.1.2 Principes généraux en matière de sûreté de fonctionnement des logiciels

Le logiciel, qui est par essence un produit immatériel, n'est pas soumis aux mêmes principes de défaillance que les matériaux plus classiques.

Si l'introduction d'un défaut (faute de conception, de codage, de transformations…) est un processus très aléatoire et non déterministe, le comportement d'un logiciel sur l'activation d'un défaut et l'apparition de défaillances sont eux **déterministes** : toute activation d'un logiciel dans des conditions de sollicitation pour lesquelles il n'est pas correctement conçu entraînera l'occurrence de la même défaillance.

**Le caractère probabiliste d'une défaillance déterministe n'est alors plus lié à la probabilité d'occurrence d'un défaut mais uniquement lié à la probabilité de sollicitation du logiciel pour un cas de fonctionnement (environnement + entrées) qui n'apporte pas une réponse satisfaisante.**

La perception du niveau de sûreté de fonctionnement d'un même logiciel peut donc être totalement différente selon le mode de sollicitation mis en œuvre : ce sont les postulats de fonctionnement du système. Le système est en fait un sous système fonctionnant même en mode dégradé dans un environnement qu'il ne maîtrise pas.

Les démarches de construction et de validation de la sûreté de fonctionnement d'un logiciel doivent aujourd'hui viser une couverture aussi grande que possible des modes de fonctionnement, tant dans l'identification des comportements attendus que dans la mise en place des mécanismes appropriés dans la démonstration du bon fonctionnement du logiciel.

Aussi, il devient nécessaire aujourd'hui [Gartner, 2009] de mettre en œuvre des techniques spécifiques pour :

- identifier les besoins en matière de sûreté de fonctionnement (recensement le plus exhaustif possible des modes fonctionnement en mode nominal et en présence de défauts : matériel, initialisation, communication, opérateur, environnement) ;
- éviter l'introduction de défauts dans le processus de conception et de réalisation du logiciel ;
- mettre en place des mécanismes de robustesse et de tolérance de défauts compatibles avec les contraintes de continuité de mission ;

## 5.1.2 Allgemeine Grundsätze der Softwaresicherheit

Software ist von Natur aus immateriell. Sie unterliegt nicht denselben Versagensmechanismen wie physisches Material.

Das Verursachen eines Fehlers (Konzeptionsfehler, Codierfehler, Umsetzungsfehler…) ist ein stark zufallsbedingter und nicht deterministischer Vorgang.

Das Verhalten einer Software bei Aktivierung eines Fehlers und Auftreten von Ausfällen ist **deterministisch**: jede Aktivierung von Software unter Betriebsbedingungen, für die sie nicht entwickelt worden ist, wird das Auftreten desselben Versagens bewirken.

**Der Wahrscheinlichkeitscharakter eines deterministischen Versagens steht dann nicht mehr mit der Wahrscheinlichkeit des Auftretens eines Fehlers in Zusammenhang, sondern nur mit der Wahrscheinlichkeit der Beanspruchung der Software (Umwelt + Eingänge) unter Bedingungen, die keine zufriedenstellenden Ergebnisse liefern.**

Die Wahrnehmung des Sicherheitsniveaus ein und derselben Software kann also gänzlich verschieden sein, je nach Beanspruchungsart. Dies sind die Anforderungen an die Funktionalität des Systems. Das System ist im Grunde genommen ein Teilsystem, das selbst in der Rückfallebene in einem von ihm nicht beherrschten Umfeld funktioniert.

Das Vorgehen bei der Entwicklung und der Sicherheitsüberprüfung einer Software muss heute eine möglichst große Abdeckung der unterschiedlichen Beanspruchungsfälle erreichen, sowohl bei der Identifizierung des erwarteten Verhaltens als auch beim Einsatz geeigneter Überprüfungsmechanismen für die Systemsicherheit.

Es ist heute nötig, spezifische Techniken einzusetzen um:

- die Bedürfnisse hinsichtlich der Sicherheit zu identifizieren (möglichst vollständige Erfassung der Funktionsweisen im Normalbetrieb und beim Auftreten von Fehlern: Hardware, Initialisierung, Bedienpersonal, Umwelt).
- Fehler bei der Konzeption und Umsetzung der Software zu vermeiden.
- Mechanismen für die Robustheit und Fehlertoleranz einzuführen, die mit den Randbedingungen der Aufrechterhaltung des Betriebs kompatibel sind.

- vérifier, dans le maximum de cas de sollicitation possibles, le comportement du logiciel (y compris en présence de défauts supposés) ;
- évaluer le niveau de sûreté de fonctionnement obtenu par un logiciel et donc du système dans lequel il est intégré.

Ces mesures sont aujourd'hui modulées en fonction du niveau des exigences et des ressources disponibles (CPU, mémoire, coûts, délais…).

## 5.1.3 Exigences en matière de sûreté de fonctionnement du logiciel

Pour obtenir les exigences de sûreté de fonctionnement pour un logiciel, les équipes de sûreté de conception du système procèdent à des analyses préliminaires de risques (permettant d'identifier *a priori* les exigences au niveau du système) et, en fonction des choix d'architecture, à effectuer une allocation des exigences de sûreté de fonctionnement pour chacun des éléments constitutifs de l'architecture.

Classiquement, l'expression des besoins peut prendre plusieurs formes :

- exigences de sécurité :
  - liste des événements redoutés pour lesquels on attend une démonstration de la réduction du risque en dessous d'un niveau jugé acceptable ;
  - modes dégradés, comportement sur défaillance : qui vont traduire la nécessité de protéger le logiciel et donc le système vis-à-vis d'une défaillance unique et de démontrer la capacité du logiciel à conserver un comportement opérationnel sûr ;
- exigence de fiabilité principalement en termes de MTTF (Mean Time To Failure) ou de taux de défaillance objectif demandé à un logiciel. Il s'agit d'exigences limitées à des logiciels non sécuritaires car elles ne peuvent être démontrées pour des valeurs élevées de MTTF ;
- exigences de maintenabilité (en général exprimées en termes de délais de restauration d'un fonctionnement nominal ou de délais pour corriger des défauts bloquants).

Classiquement, les attendus en matière de construction, de vérification et d'évaluation du niveau de sûreté de fonctionnement varient.

- für möglichst viele Beanspruchungsfälle das Softwareverhalten zu prüfen (einschließlich des Auftretens erwarteter Fehler).
- das Niveau der Funktionssicherheit der Software und somit des Systems in dem sie betrieben wird abzuschätzen.

Diese Maßnahmen werden heute dem Niveau der Anforderungen und der verfügbaren Mittel angepasst (CPU, Speicher, Kosten, Fristen…).

## 5.1.3 Forderungen für die Funktionssicherheit von Software

Um die Sicherheitsanforderungen an eine Software zu bestimmen, führen die mit der Sicherheit beauftragten Entwicklungsteams vorläufige Risikoanalysen durch (die es erlauben, die Anforderungen an das System im Voraus zu identifizieren). Sie ordnen, entsprechend der Architektur, jedem wesentlichen Bestandteil des Systems eine Sicherheitsanforderung und -stufe zu.

Üblicherweise nimmt dieser Ausdruck der Bedürfnisse mehrere Formen an:

- Sicherheitsanforderungen:
  - Liste der befürchteten Ereignisse, für die man einen Nachweis der Risikoreduzierung unter ein für annehmbar gehaltenes Niveau erwartet
  - Rückfallebene, Verhalten bei Versagen: die Software und somit das System müssen gegen ein einmaliges Versagen geschützt werden, und es muss nachgewiesen werden, dass das System ein sicheres Betriebsverhalten beibehalten kann.
- Zuverlässigkeitsanforderung hauptsächlich in Bezug auf die MTTF (Mean Time To Failure) oder objektive Ausfallrate, die von einer Software verlangt wird: es handelt sich um Anforderungen, die auf eine nicht sicherheitsrelevante Software begrenzt sind, denn die Anforderungen können für hohe MTTF-Werte nicht nachgewiesen werden.
- Anforderungen an die Instandhaltbarkeit (im Allgemeinen ausgedrückt in Bezug auf die Dauer der Wiederherstellung des Normalbetriebs oder der Dauer der Korrektur eines blockierenden Fehlers).

Üblicherweise variieren die Erwartungen hinsichtlich der Konstruktion, der Prüfung und der Bewertung des Niveaus der Funktionssicherheit.

## 5.1.4 Construction de la sûreté de fonctionnement des logiciels

La construction de la sûreté de fonctionnement d'un logiciel et par extension d'un système programmé vise à conférer à ce système le niveau de sûreté de fonctionnement requis. Pour cela les solutions envisagées reposent sur un ensemble de « pratiques » à mettre en œuvre et qui doivent permettre d'atteindre un niveau plus ou moins élevé de sûreté de fonctionnement qu'il sera cependant difficile de qualifier et a fortiori de quantifier. Classiquement, les différents volets qui permettent de construire la sûreté de fonctionnement sont les suivants :

- mise en œuvre de méthodes de type « analyse des défaillances » qui permettent d'identifier à partir des événements redoutés donnés en entrée de la spécification du logiciel, les fonctions, modules ou composants le plus critiques ;
- choix de l'architecture selon le niveau de sûreté de fonctionnement et de tolérance aux fautes visées (architectures redondantes…) ;
- définition et implémentation de mécanismes de sécurisation : programmation défensive pour réduire le risque de propagation d'erreurs internes du logiciel ou d'erreurs du matériels ;
- utilisation de règles de développement orientées sûreté de fonctionnement : règles de conception, de structuration, d'utilisation des ressources, de codage… qui permettent d'éviter de reproduire des erreurs survenues dans le passé.

Les différentes normes qui traitent de sûreté de fonctionnement des systèmes programmés définissent des grilles d'exigences qui modulent l'utilisation de ces différentes approches selon le niveau de fonctionnement visé [EN50126, 2000] [DO178B, 1992] [L108/4, 2009] [2004L0049]. Comme nous l'avons vu au chapitre 3, les obligations de moyens traduites par ces « modulations » n'ont pas été en mesure de se traduire dans les faits une obligation de résultat.

Aujourd'hui, l'utilisation d'approches dites formelles pour la spécification, la conception et le développement est considérée comme un moyen efficace de construire un logiciel sûr de fonctionnement [Clearsy, 2006] [Gartner, 2009]. Néanmoins, la complexité des langages de modélisation, la difficulté de répondre à tout type d'exigences et la non disponibilité sur le marché d'outils peu chers et ergonomiques de validation formelle et de génération automatique de code sont considérés encore comme ne permettant pas une application industrielle. [Desroche, 2003] [Goeg, 2003] [IMdR, 2005] [Jahnel, 2000] [Méry, 2006] [Tarnai, 2006] [Tarnai, 2009].

## 5.1.4 Konstruktion der Funktionssicherheit von Software

Die Konstruktion der Softwaresicherheit und somit die Sicherheit eines programmierten Systems zielt darauf ab, diesem System das Niveau der erforderlichen Systemsicherheit zu geben. Dies beruht auf einer Gesamtheit von „Praktiken", die umgesetzt werden. Diese müssen es erlauben, ein mehr oder weniger hohes Sicherheitsniveau zu erreichen. Es ist jedoch schwierig, diese Funktionssicherheit zu qualifizieren und mehr noch, zu quantifizieren. Die verschiedenen Elemente, die es üblicherweise erlauben, die Funktionssicherheit herzustellen sind folgende:

- Erstellen einer Fehlerbaumanalyse. Sie erlaubt es, von den befürchteten Ereignissen ausgehend, die als Eingabe der Spezifikationen dienen, die kritischsten Funktionen, Module oder Bestandteile zu identifizieren.
- Architekturwahl entsprechend dem angestrebten Sicherheitsniveau und der angestrebten Fehlertoleranz
- Definition und Implementierung von Sicherungsmechanismen: Abwehrmaßnahmen, um das Risiko der Verbreitung interner Software- oder Gerätefehler zu reduzieren
- Einhaltung von Entwicklungsgrundsätzen, die systemsicherheitsorientiert sind: Konzeptionsregeln, Strukturierungsregeln, Regeln der Ressourcenbenutzung, Regeln beim Codieren, usw. die es erlauben, in der Vergangenheit aufgetretene Fehler zu vermeiden

Die verschiedenen programmierte Systeme betreffenden Normen definieren Anforderungstabellen, die die Benutzung dieser verschiedenen Konzepte je nach angestrebtem Sicherheitsniveau anpassen. [EN50126, 2000] [DO178B, 1992] [L108/4, 2009] [2004L0049]. Wie in Kapitel 3 beschrieben, sind die Auflagen bezüglich der eingesetzten Mittel, die durch diese „Modulationen" beschrieben werden, nicht in der Lage, Auflagen hinsichtlich des Ergebnisses auszudrücken.

Heute wird die Benutzung formaler Methoden für die Spezifizierung, die Konzeption und die Entwicklung als ein wirksames Mittel angesehen, eine sichere Software herzustellen. [Clearsy, 2006] [Gartner, 2009]. Trotzdem führen die Komplexität der Modellierungssprachen, die Schwierigkeit, allen Forderungen nachzukommen und die Nichtverfügbarkeit ergonomischer, marktverfügbarer und kostengünstiger Anwendungsprogramme für formale Überprüfung und automatische Codeerzeugung dazu, dass diese Methoden für die industrielle Anwendung als noch nicht ausgereift angesehen werden. [Desroche, 2003] [Goeg, 2003] [IMdR, 2005] [Jahnel, 2000] [Méry, 2006] [Tarnai, 2006] [Tarnai, 2009]

### 5.1.5 Vérification de la sûreté de fonctionnement du logiciel

Les normes [2004L0049] [2004R0881] [L108/4,2009] imposent la rédaction d'un «plan de vérification» et d'un «plan de validation». Pour la vérification, il s'agit de mettre en œuvre des relectures systématiques de toute la documentation produite, qu'elle soit «projet» ou spécifiquement «sûreté de fonctionnement». Pour la vérification de la documentation «produit», les relectures doivent s'assurer que les documents sont lisibles, qu'ils sont sans défauts et que les exigences normatives sont respectées. Pour la validation de la sûreté de fonctionnement, il s'agit principalement de s'assurer que les exigences de sûreté de fonctionnement sont atteintes et que l'ensemble des cas test passés confère au logiciel la confiance attendue [Kahn, 2008]. Cette « validation » est en principe effectuée tout au long de la branche remontante du cycle de vie en V. Les techniques de validation peuvent être manuelles ou automatisées. Elles dépendent du niveau de sûreté de fonctionnement alloué au logiciel, du temps alloué, de sa complexité

Les normes s'intéressent particulièrement à l'évaluation de la couverture de test. Les normes considèrent qu'une couverture exhaustive est inenvisageable et qu'il s'agit uniquement de mettre en œuvre un ensemble de cas de test qui garantisse néanmoins que le risque résiduel est acceptable [EN50128, 2001]. L'acceptation de la validation de la sûreté de fonctionnement du logiciel ne peut être prononcée que lorsqu'on a procédé, sur une version embarquée du logiciel et dans des considérations d'environnement et d'utilisation « proche » de la réalité, à un ensemble d'essais jugés satisfaisants pour corroborer ceux effectués lors des phases précédentes de test unitaire, d'intégration et de validation du logiciel.

La démonstration de l'atteinte d'un niveau de sûreté de fonctionnement satisfaisant pour un logiciel doit être obtenue pour chaque version livrée du logiciel, ce qui nécessite le plus souvent d'automatiser, autant que possible, le passage des cas de test et la mesure du taux de couverture (ou de définir une stratégie de non régression).

**Ainsi, pour les plus hauts niveaux d'exigence, les coûts et délais requis pour chaque évolution fonctionnelle d'un système programmé sont très élevés, sans qu'ils puissent donner une garantie absolue.**

### 5.1.5 Prüfung der Funktionssicherheit von Software

Die Normen [2004L0049] [2004R0881] [L108/4,2009] verlangen das Verfassen eines „Prüfplans" und eines „Validierungsplans". Bei der Prüfung handelt es sich um eine systematische zweite Lektüre der gesamten erstellten Dokumentation, ob es sich nun um das Projekt an sich oder spezifisch um die Funktionssicherheit handelt. Bei der Prüfung der „Produktdokumentation" muss die Zweitlektüre gewährleisten, dass die Dokumente lesbar und fehlerfrei sind und dass die normativen Anforderungen respektiert werden. Bei der Bewertung der Sicherheit handelt es sich hauptsächlich darum, zu gewährleisten, dass die Sicherheitsanforderungen erreicht werden, und dass alle durchgeführten Testfälle der Software das erwartete Vertrauen verleihen [Kahn, 2008]. Diese Bewertung wird im Prinzip während des aufsteigenden Asts des V-Lebenszyklus durchgeführt. Die Bewertungstechniken können manuell oder automatisch sein. Sie hängen vom Niveau der Funktionssicherheit der Software, von der verfügbaren Zeit und von der Komplexität der Software ab.
Die Normen betreffen insbesondere die Bewertung der Abdeckung durch den Test. Sie gehen davon aus, dass eine komplette Abdeckung nicht in Erwägung gezogen werden kann. Es sind lediglich Testfälle zu realisieren, um zu gewährleisten, dass das Restrisiko annehmbar ist [EN50128, 2001]. Die Validierung der Softwarefunktionssicherheit kann erst dann akzeptiert werden, wenn mit einer onboard Variante der Software unter „wirklichkeitsnahen" Bedingungen eine Reihe von zufriedenstellenden Versuchen durchgeführt wurde, um die bereits früher durchgeführten Versuche (Einzeltests, Integrationstests und Softwarevalidierung) zu bestätigen.

Der Nachweis einer bestimmten, für eine Software zufriedenstellenden Funktionssicherheit ist für jede gelieferte Softwareversion notwendig. Deswegen sollten die Testfälle sowie die Messung des Abdeckungsgrades möglichst automatisiert sein (bzw. die Strategie gegen Auftreten neuer Fehler muss definiert sein).

**Für höhere Anforderungsniveaus sind also die für jede funktionelle Weiterentwicklung eines programmierten Systems notwendigen Kosten und Fristen sehr hoch, ohne dass es deshalb eine absolute Gewährleistung gibt.**

## 5.1.6 Évaluation de la fiabilité des logiciels

Pour les logiciels peu critiques, une évaluation de la fiabilité des logiciels doit démontrer une exigence quantifiée de MTTF ou de taux de défaillance. Pour ce faire, il faut procéder à un relevé du comportement du logiciel pendant un temps de fonctionnement suffisant pour démontrer que l'objectif est tenu. Ce temps dépend de l'objectif initial fixé et du nombre de défaillance que va présenter le logiciel évalué. Moins le logiciel est fiable au départ, plus le temps de validation pour atteindre l'objectif est long.

Comme les défauts mis en évidence lors du test sont corrigés, des modèles dits « *de croissance de fiabilité du logiciel* » [Vallée, 1998] [Desroche, 2003] sont classiquement utilisés. Ces modèles font principalement deux hypothèses :

- la correction des défauts est immédiate, c'est-à-dire qu'une défaillance due à un défaut donné n'apparaît qu'une fois ;
- cette correction est parfaite, c'est-à-dire qu'elle ne crée pas de nouveaux défauts dans le code.

La véracité de ces hypothèses est discutable : preuve en est, les efforts importants que consacrent les industriels aux «tests de non régression», censés les protéger contre l'introduction de nouveaux défauts lors de toute manipulation de code… [Legof, 2009] [SNCF, 2005] [Bombardier, 2005] [01Info, 2001] [CALIFE, 2001]

Notons que les modèles de fiabilité héritent des limitations propres à l'utilisation du retour d'expérience pour valider l'obtention d'un objectif de fiabilité, à savoir : les heures de fonctionnement à cumuler dans ce retour d'expérience sont inversement proportionnelles à la valeur du taux de défaillance à estimer. Cela signifie que pour évaluer un taux de défaillance de $10^{-5}$ par heure, il faut cumuler un temps d'expérience sans défaillance d'au moins 100000 heures et même 400000 si l'on veut avoir une bonne confiance dans l'évaluation![34]. Or 400.000 heures représentent 45 ans de fonctionnement à plein temps !

**De ce qui précède, il découle que, dans le domaine de fiabilité du logiciel, il est illusoire d'imposer d'évaluer une exigence inférieure à $10^{-3}$ défaillance par heure.**

## 5.1.6 Bewertung der Zuverlässigkeit von Software

Für unkritische Software beinhaltet die Zuverlässigkeitsbewertung den Nachweis einer quantifizierten MTTF bzw. einer quantifizierten Ausfallrate. Das Softwareverhalten muss lange genug beobachtet werden, um nachzuweisen, dass die Anforderungen auch eingehalten werden. Diese Dauer hängt von den ursprünglichen Anforderungen und von der Anzahl der Ausfälle der zu bewertenden Software ab. Je weniger zuverlässig die Software zu Beginn ist, desto länger ist die Validierungsdauer zur Einhaltung der Anforderungen.

Da die während des Testes aufgezeichneten Fehler korrigiert werden, verwendet man üblicherweise sogenannte Modelle der „*Softwarezuverlässigkeitssteigerung*" [Vallée, 1998] [Desroche, 2003]. Diese beruhen im Wesentlichen auf zwei Annahmen:

- sofortige Fehlerbehebung, so dass ein Ausfall aufgrund eines bestimmten Fehlers nur einmal auftritt
- perfekte Fehlerbehebung, so dass im Code keine neuen Fehler erzeugt werden.

Die Richtigkeit beider Annahmen ist umstritten: ein Beweis dafür sind die großen Anstrengungen der Industrie bei den „Nichtverschlechterungstests" zum Schutz gegen die Einführung neuer Fehler durch Codeänderungen [Legof, 2009] [SNCF, 2005] [Bombardier, 2005] [01Info, 2001] [CALIFE, 2001].

Zuverlässigkeitsmodelle haben, da sie auf Erfahrungswerten basieren, folgende Grenzen bei der Bewertung eines Zuverlässigkeitsniveaus: die gesamten Betriebsstunden sind umgekehrt proportional zu der zu schätzenden Ausfallrate. Um eine Ausfallrate von $10^{-5}$ pro Stunde zu bewerten, braucht man mindestens 100.000 oder gar 400.000 ausfallfreie Stunden, um der Schätzung vertrauen zu können.![35]. Aber 400.000 Stunden sind 45 Betriebsjahre!

**Daraus ergibt sich, dass es im Hinblick auf die Softwarezuverlässigkeit illusorisch ist, die Bewertung einer Zuverlässigkeit unter $10^{-3}$ Ausfällen pro Stunde zu fordern.**

[34] Eventuellement les systèmes peuvent être testés en parallèle pour réduire ce délai

[35] Der Parallele Test von vielen Anlagen ist möglich um diese Dauer zu reduzieren

## 5.1.7 Formalisation de la sûreté de fonctionnement du logiciel

La mise en place d'une démarche de sûreté de fonctionnement lors du développement d'un logiciel (et d'un système programmé) doit pouvoir être lisible et claire. Elle doit pouvoir être formalisée que ce soit pour :

- définir et faire approuver les dispositions qui sont prises pour atteindre le résultat acceptable attendu ;
- donner confiance dans l'obtention effective de ce résultat.

Le plan de sûreté de fonctionnement doit définir les principaux éléments de la démarche mise en œuvre :

- organisation de la sûreté de fonctionnement : rôle des acteurs…
- rappel des exigences de sûreté de fonctionnement : exigences globales de sûreté de fonctionnement du système complet, classes de criticité des fonctions (niveau de SIL [EN50126, 2000] définis par les normes CEI 61508, EN 50128…) ;
- principes de démarche de sûreté de fonctionnement mis en œuvre : choix structurant d'architecture, positionnement des activités de sûreté de fonctionnement dans le développement…
- détail des activités de sûreté de fonctionnement ;
- gestion de la documentation : statut et gestion des produits, principes de diffusion…
- méthodes et outils pour la construction de la sûreté de fonctionnement ;
- interface avec les autres composantes du système : aspects logiciel, matériel, système…
- planification des activités…
- suivi d'application du plan de sûreté de fonctionnement…

Le plan de sûreté de fonctionnement logiciel est livré au client, il est approuvé par ce dernier et devient un engagement contractuel du fournisseur.

Le dossier de sûreté de fonctionnement doit permettre au fournisseur de montrer qu'il a effectivement obtenu les résultats attendus et correctement mis en œuvre le plan de sûreté de fonctionnement approuvé par le client.

## 5.1.7 Planung und Dokumentation der der Softwarefunktionssicherheit

Bei der Entwicklung von Software (eines programmierten Systems) muss die Vorgehensweise bezüglich der Funktionssicherheit verständlich und klar sein. Sie muss aus folgenden Gründen formal gestaltet werden:

- Definition und Genehmigung der Vorgehensweisen, die eingesetzt werden, um das angestrebte annehmbare Ergebnis zu erreichen
- Vertrauen darauf, dieses Ergebnis erreicht zu haben.

Der Funktionssicherheitsplan muss die wesentlichen Elemente der Vorgehensweise definieren:

- Organisation der Funktionssicherheit : Rolle der beteiligten Personen
- Erinnerung an die Funktionssicherheitsanforderungen: globale Funktionssicherheitsanforderungen an das komplette System, in den Normen IEC61508, EN50128, usw. definierte Funktionkritizitätsklassen SIL [EN50126, 2000]
- Grundsätze der umgesetzten Funktionssicherheit: Architekturwahl, Positionieren der Funktionssicherheitsmaßnahmen während der Entwicklung, usw.
- Details der Funktionssicherheitsmaßnahmen
- Dokumentationsverwaltung : Status und Verwaltung der Produkte, Grundsätze der Informationsverteilung, usw.
- Verfahren und Tools zur Schaffung der Funktionssicherheit
- Schnittstelle mit den anderen Systemkomponenten : Softwareaspekte, Hardwareaspekte, Systemaspekte, usw.
- Planung der Maßnahmen
- Begleitung der Umsetzung des Funktionssicherheitsplans…

Der Funktionssicherheitsplan für Software wird dem Kunden übergeben; dieser genehmigt ihn, bevor es zu einer vertraglichen Verpflichtung des Lieferanten kommt.

Das Funktionssicherheitsdokument muss es dem Lieferanten erlauben, nachzuweisen, dass er die angestrebten Ergebnisse erreicht und den Betriebssicherheitsplan korrekt umgesetzt hat.

Ce dossier doit aussi permettre de donner confiance au client quant au niveau réel de sûreté de fonctionnement obtenu. En ce sens, il ne constitue pas seulement une concaténation de résultats divers mais doit présenter les analyses effectuées par les personnes en charge de la sûreté de fonctionnement du logiciel chez le fournisseur et les résultats obtenus en regard des différentes dispositions du plan de sûreté de fonctionnement, sur la base desquels le fournisseur juge que le niveau de sûreté de fonctionnement obtenu est satisfaisant.

Le dossier de sûreté de fonctionnement est une fourniture importante pour «**donner confiance**» dans le niveau de sûreté de fonctionnement obtenu. Sa rédaction ne donne en fait aucune certitude sur le résultat obtenu mais seulement sur les moyens effectivement mis en œuvre à cet effet.

## 5.1.8 Remarques

Si les normes en matière de sûreté de fonctionnement convergent globalement vers un référentiel méthodologique de développement cohérent, elles présentent encore, selon les enjeux et les capacités technico-économiques du domaine, des différences non négligeables dans les exigences qu'elles prescrivent : un logiciel critique dans l'automobile ne bénéficiera pas d'autant de précautions qu'un logiciel aéronautique ou ferroviaire [liste des références DGA par exemple].

Si le niveau de sûreté de fonctionnement, et en particulier le niveau « acceptable » d'occurrence de défaillance critique, est souvent utilisé pour déterminer la classe de criticité du logiciel à produire et, en conséquence, l'effort en matière de méthodes et de techniques de sûreté de fonctionnement nécessaire pour réduire le risque résiduel à un niveau jugé acceptable, rien ne permet véritablement de prouver le niveau réellement obtenu car les estimations quantifiées prévisionnelles s'appuient sur le taux de défaillance du matériel.

**Seule l'utilisation de méthodes formelles, considérées aujourd'hui encore comme complexes et de ce fait rarement utilisées, permettrait d'atteindre ce résultat.**

Les normes légitiment ainsi l'existence dans les systèmes mis en exploitation de situations à risque avec un niveau d'occurrence jugés «acceptables».

L'efficacité des normes repose en fait essentiellement sur le niveau d'indépendance élevé entre valideurs et concepteurs.

Ferner gibt es dem Kunden Vertrauen in die erzielte Funktionssicherheit: Das Dokument stellt also nicht nur eine Zusammenfassung der verschiedenen Ergebnisse dar, sondern es muss auch die Analysen der vom Lieferanten für die Funktionssicherheit beauftragten Personen enthalten, sowie die Ereignisse bezüglich der verschieden Maßnahmen des Funktionssicherheitsplans, auf deren Basis der Lieferant folgert, dass die Funktionssicherheit zufriedenstellend ist.

Das Funktionssicherheitsdokument ist ein wichtiger Teil der Lieferung, denn es **erzeugt Vertrauen** in die erzielte Funktionssicherheit. Aber eigentlich gibt dieses Dokument keine Gewissheit über das erzielte Ergebnis, sondern nur über die eingesetzten Mittel.

## 5.1.8 Anmerkungen

Selbst wenn die Funktionssicherheitsnormen sich im Allgemeinen alle in Richtung eines methodologischen Bezugswertes für eine kohärente Entwicklung verändern, gibt es, je nach dem, was auf dem Spiel steht, und je nach den technisch-wirtschaftlichen Möglichkeiten des Fachgebiets, bedeutende Unterschiede bei den vorgeschriebenen Forderungen: bei einer kritischen Software werden im Automobilbereich nicht so viele Vorsichtsmassnahmen getroffen wie im Luftfahrt- oder Bahnbereich [liste des références].

Das Niveau der Funktionssicherheit (und insbesondere die „akzeptable" Häufigkeit kritischer Ausfälle) dient oft der Bestimmung der Kritizität der entwickelten Software und folglich der Definition der Verfahrens- und der Funktionssicherheitstechniken, um das Restrisiko auf ein als „annehmbar" geltendes Niveau zu reduzieren. Es ist jedoch nicht möglich, wirklich nachzuweisen, welches Niveau tatsächlich erreicht wurde, da sich quantifizierte Vorhersagen auf die Hardwareausfallrate stützen.

**Lediglich die heute immer noch als komplex geltenden und selten verwendeten formalen Verfahren erlauben es, zu diesem Ergebnis zu gelangen.**

Auf diese Weise legitimieren die Normen das Vorhandensein von Risikosituationen – in betriebenen Systemen – mit einer als „annehmbar" geltenden Häufigkeit.

Die Wirksamkeit der Normen beruht im Wesentlichen auf der hohen Unabhängigkeit zwischen der Systementwicklung und der Systemvalidierung.

Les normes {EN50126, 2000] [EN50128, 2001] [EN50129, 2003] [L108/4, 2009] reposent sur une identification «quasi exhaustive» des événements redoutés du système. Classiquement celle-ci est confiée aux experts sûreté de fonctionnement du projet de développement (les personnes du comment), et non aux gens de métiers (les personnes du pourquoi). Ainsi les aspects gestion des modes dégradés, changement de profils de missions, liens avec les procédures en vigueur dans le métier… sont autant de sources avérées de d'erreurs de spécification et d'oublis d'événement redoutés.

L'application des normes paraît peu adaptée aux systèmes fortement paramétrables (comme les postes d'aiguillage). Elles ne couvrent en effet essentiellement le logiciel exécutable et non les paramètres fonctionnels (même si les normes abordent les paramètres).

En toute rigueur il serait nécessaire de considérer tout nouvel ensemble «logiciel de base + paramètres site» comme une nouvelle version logicielle à revalider en tant que tel. Classiquement, le logiciel de base est validé avec un jeu de paramètres (celui de la première cible), les jeux de paramètres suivants sont testés en considérant que le logiciel de base est correct.

Les normes font implicitement la supposition d'un lien de proportionnalité entre niveau d'exigence de sûreté de fonctionnement et coûts de développement (modulation des exigences en moyens et méthodes à mettre en œuvre…).

**Seule l'utilisation outillée de méthodes formelles, appliquées par des experts du métier de l'application considérée (formation par compagnonnage, équipe projet avec un mainteneur et un exploitant) sur les systèmes programmés paramétrés, permettrait de prouver, indépendamment du niveau de SIL, que le logiciel est « correct » dans son contexte d'usage.**

Die Normen [EN50126, 2000] [EN50128, 2001] [EN50129, 2003] [L108/4, 2009] beruhen auf der „quasi-kompletten" Identifizierung der befürchteten Ereignisse des Systems. Üblicherweise obliegt diese Identifizierung den Funktionssicherheitsexperten des Entwicklungsprojektes (denjenigen, die das „Wie" bestimmen), und nicht den Fachmännern (denjenigen, die das „Warum" bestimmen). Gesichtspunkte wie Management der Rückfallebene, Änderung der Aufgabenprofile, Verbindung mit den im Fachgebiet geltenden Vorschriften, usw. sind nachweislich häufige Ursachen für Spezifikationsfehler und das Vergessen befürchteter Ereignisse.

Die Anwendung der Normen scheint im Falle von stark parametrierbaren Systemen (wie Stellwerken) nicht sehr angebracht zu sein. In der Tat betreffen diese Normen im Wesentlichen die ausführbare Software und nicht die funktionellen Parameter (selbst wenn Parameter in den Normen erwähnt werden).

Will man präzise sein, so muss man jede neue Kombination „Grundsoftware und Standortparameter" als eine neue Softwareversion betrachten und erneut validieren. Üblicherweise wird die Grundsoftware mit einem Satz Parametern (denjenigen der ersten Zielanwendung) validiert, und die nächsten Parametersätze werden unter der Annahme getestet, dass die Basissoftware korrekt ist.

Die Normen gehen implizit davon aus, dass es eine Proportionalität zwischen der Funktionssicherheitsanforderung und den Entwicklungskosten (Anpassung der Anforderungen je nach eingesetzenden Mitteln und Verfahren…) gibt.

**Nur die Verwendung formaler Verfahren durch Fachleute (praktische Ausbildung, Projektteam mit Wartungs- und Betriebspersonal) auf dem Gebiet der Anwendung programmierter und parametrierter Systeme kann unabhängig vom SIL beweisen, dass die Software für ihren Verwendungszweck „korrekt" ist.**

## 5.2 Liens des méthodes formelles avec la sûreté de fonctionnement

Les méthodes formelles sont une des techniques envisagées depuis maintenant près de 20 ans pour traiter la problématique spécifique de Sûreté de Fonctionnement des logiciels. Certaines méthodes formelles sont bien plus anciennes mais ne s'appliquaient alors aux enclenchements des postes d'aiguillage mécaniques [Pichon, 1886] [Descubes, 1898] (cf. Annexe C). Évoquons le cadre général des différentes techniques utilisées en Sûreté de Fonctionnement logicielle, pour mieux situer la plus value spécifique constituée par l'apport de ces méthodes formelles.

### 5.2.1 Méthodes d'Analyse de la Fiabilité Logicielle

Ces différentes méthodes peuvent être regroupées en plusieurs familles :

- les techniques de fiabilisation apparues dans le cadre de l'évolution des différentes méthodes de développement ;
- les AEEL (Analyses d'Effets d'Erreurs Logicielles), transposition des AMDEC aux systèmes logiciels ;
- les autres méthodes, techniques et modèles classiques de Sûreté de Fonctionnement habituellement appliqués aux systèmes Matériel et transposés aux systèmes logiciels ;
- les modèles de taux de défaillances logiciels (croissances de fiabilité suites aux actions de correction d'anomalies) ;
- les **méthodes formelles** dont nous évoquerons les caractéristiques spécifiques par rapport aux techniques citées précédemment.

#### 5.2.1.1 Les AEEL

Les AEEL présentent les avantages suivants :

- elles permettent de produire un retour constructif du point de vue de l'aide à la conception, et de mettre en évidence des améliorations des applications développées ;
- elles présentent un aspect pédagogique et constructif : elles ne se limitent pas à évaluer le niveau de fiabilité intrinsèque d'une application, mais elles mettent en évidence naturellement des actions concrètes d'amélioration.

## 5.2 Verbindung zwischen formalen Methoden und Funktionssicherheit

Formale Methoden sind seit etwa 20 Jahren bekannte Techniken, um die spezifische Sicherheitsproblematik von Software zu behandeln. Bestimmte formale Methoden sind sehr viel älter. Sie fanden zum Beispiel bei dem Verschließen mechanischer Stellwerke Anwendung [Pichon, 1886] [Descubes, 1898] (vgl. Anhang C). Im Folgenden wird der allgemeine Rahmen verschiedener Techniken beschrieben, die auf dem Gebiet der Funktionssicherheit von Systemen verwendet werden. Dies erlaubt es, den Vorteil formaler Methoden besser darzustellen.

### 5.2.1 Analysemethoden der Softwarezuverlässigkeit

Die verschiedenen Verfahren können in mehreren Familien zusammengefasst werden:

- Techniken der Zuverlässigkeitsverbesserung die im Rahmen der verschiedenen Entwicklungsmethoden entstanden sind
- SFMEA (Software failure modes and Effekts Analysis - Analyse der Auswirkungen von Softwarefehlern), Umsetzung der FMECA (Fehlermöglichkeits- und Einflussanalyse oder kurz Auswirkungsanalyse) für Softwaresysteme
- Andere Verfahren, Techniken und klassische Modelle der Funktionssicherheit, die gewöhnlich auf Hardwaresysteme angewendet werden und die für Softwaresysteme angepasst werden
- Modelle für Softwareausfallraten (Zuverlässigkeitsverbesserung durch Fehlerbehebung)
- **formale Verfahren**, deren spezifische Eigenschaften mit denen der aufgeführten Techniken verglichen werden.

#### 5.2.1.1 Die SFMEA

Die SFMEA hat folgende Vorteile:

- Sie erlaubt es, hinsichtlich der Konzeptionsunterstützung eine konstruktive Rückmelddung zu erstellen und Verbesserungen der entwickelten Anwendungen hervorzuheben.
- Sie hat einen pädagogischen und einen konstruktiven Aspekt: sie beschränkt sich nicht darauf, das eigentliche Zuverlässigkeitsniveau einer Anwendung abzuschätzen, sondern sie zeigt von sich aus konkrete Verbesserungsmaßnahmen auf.

Les AEEL permettent de mettre en évidence :

- des aspects algorithmiques qui ont été négligés, en ne couvrant pas suffisamment des cas limites susceptibles de caractériser des configurations de données non prévues ;
- des possibilités de vérifications, de contrôles de cohérence complémentaires permettant de rendre des traitements plus robustes aux cas particuliers, erreurs de calculs ou de données ;

**Les AEEL ne sont pas applicables aux logiciels complets des modules informatiques de sécurité (modules d'enclenchement dans le vocable ferroviaire) compte tenu de leur taille. Les AEEL sont par contre utilisables pour des logiciels critiques particuliers, généralement indépendants du fonctionnel.**

### 5.2.1.2 Les approches de calcul de taux de défaillance des logiciels

Bien que ces méthodes soient bien connues, elles ne se sont pas généralisées dans l'industrie [IEC 61508 standards / les normes ne le demandent pas et les industriels se limitent souvent aux normes – exe Accident PAI Meulin Liste de référence], car ce type de technique nécessite une observation rigoureuse de la statistique des temps de test et de « debugging » dès le début du développement des logiciels. De plus, ces modèles n'ont pas vraiment évolué depuis 15 ans. Ces méthodes ont tendance à être valides quant elles sont appliquées à des systèmes peu fiables. La représentativité de la mesure probabiliste assimilée à ces taux de défaillance pose problème : elle dépend tout autant de l'environnement de test et d'utilisation, que de la proportion de défauts liée à la conception du logiciel.

**Ces méthodes ne sont pas utilisables ni pour les modules d'automatismes de sécurité (module d'enclenchements dans le vocable ferroviaire) compte tenu du haut niveau de sécurité attendu (SIL4), ni pour les modules informatiques de contrôle commande (SIL2). L'application pratique de ces méthodes sur les logiciels critiques de modules d'enclenchement ne permettrait pas de retrouver une approche dite « en sûreté intrinsèque ».**

Die SFMEA erlaubt es:

- algorithmische Aspekte aufzuzeigen, die vernachlässigt worden sind, da sie nicht genügend Grenzfälle, die nicht vorgesehene Konstellationen darstellen können, betreffen,.
- Möglichkeiten zur Überprüfung und zusätzliche Kohärenzkontrollen aufzuzeigen, die es erlauben, die Datenverarbeitung gegen Spezialfälle, Rechen- und Datenfehler unempfindlicher zu machen.

**Die SFMEA ist aufgrund ihrer Größe nicht auf die vollständige Software des rechnerbasierten Sicherheitsmoduls anwendbar (Sicherungsmodul in der Eisenbahnsprache). Die SFMEA ist hingegen einsetzbar für spezielle kritische Software, die im Allgemeinen unabhängig von der funktionellen Software ist.**

### 5.2.1.2 Ansätze der Ausfallratenberechnung von Software

Obwohl diese Methoden allgemein bekannt sind, haben sie sich nicht in der Industrie verallgemeinert [Liste de références]. Diese Art von Technik erfordert in der Tat zu Beginn der Entwicklung der Software eine strenge Überwachung der Test- und Fehlerbehebungsstatistiken. Außerdem hat sich diese Methode seit 15 Jahren nicht bedeutend weiterentwickelt. Diese Methoden sind akzeptabel, wenn man sie auf wenig verlässliche Systeme anwendet. Der repräsentative Charakter der Wahrscheinlichkeitsmaße, die zu den Ausfallraten gehören, ist nicht unbedingt vorhanden: er hängt genauso von der Test- und Einsatzumgebung ab, wie vom Anteil der Konzeptionsfehler.

**Diese Methoden sind aufgrund des angestrebten Sicherheitsniveaus (SIL4) nicht auf Informatiksicherheitsmodule (Sicherungsmodul in der Eisenbahnsprache) anwendbar. Sie sind auch nicht auf Informatikkontrollmodule anwendbar (SIL2). Die praktische Anwendung von diesen Verfahren auf kritische Software von Sicherungsmodulen könnte nicht so zu einem Ansatz der „inhärenten Betriebssicherheit" Ansatz führen.**

### 5.2.1.3 Positionnement des méthodes formelles au sein de ces différentes méthodes

Que ce soient les AEEL, les méthodes reposant sur l'application à des systèmes purement logiciels des techniques d'analyse par Arbres de Défaillance ou d'Évènements, Blocs Diagrammes de Fiabilité, graphes d'états chaînes de Markov, ou encore des méthodes de calcul de taux de défaillance, on constate que l'influence exercée par la connaissance du métier de la Sûreté de Fonctionnement, du point de vue du « matériel » est encore très présente.

L'analogie «Sûreté de Fonctionnement Matérielle» - «Sûreté de Fonctionnement Logicielle» pose d'ailleurs des problèmes théoriques conduisant à contester l'utilisation même de l'expression « Fiabilité Logicielle » : en effet, le caractère aléatoire est lié au processus de mise en évidence des bugs, et non aux bugs eux mêmes dont la présence au sein du code est de nature **déterministe** [Bied, 1998]. De plus, l'impression d'aléatoire attachée aux mécanismes de mise en évidence des « bugs » est plus liée à l'ignorance des concepteurs et des utilisateurs quant à l'influence de l'environnement du logiciel sur le comportement du logiciel.

**On va constater dans ce qui suit que l'introduction des méthodes formelles est bien plus liée aux métiers associés au Génie Logiciel. Au lieu de se baser sur des analogies Hardware / Software, les méthodes formelles considèrent les logiciels comme des entités abstraites à part entière, entièrement issues de l'esprit humain, et qui sont donc susceptibles de pouvoir faire l'objet d'une mathématisation, et d'aboutir à la démonstration en absolu de propriétés ou d'invariants de comportement, de conformité à des spécifications ou d'indépendance totale vis à vis de certains aléas environnementaux.**

**L'application pratique de méthodes formelles sur les logiciels critiques de modules d'enclenchement permettrait de retrouver une approche dite « en sûreté intrinsèque » ou « fail safe »**[36] [Bied, 1998] [Bied, 2003].

### 5.2.1.3 Einordnung der formalen Verfahren

Welches Verfahren auch immer benützt wird, sei es die SFMEA, auf reinen Softwaresystemen und Fehlerbaum-/Ereignisanalysen beruhende Verfahren, Zuverlässigkeitsblockdiagramme, Zustandsgraphen/Markovketten oder Verfahren zur Berechnung der Ausfallraten, so ist der Einfluss der Fachkenntnisse (Funktionssicherheitskenntnisse bezüglich der „Hardware“) noch sehr präsent.

Die scheinbare Analogie zwischen „materieller Funktionssicherheit“ und „Softwarefunktionssicherheit“ führt zu theoretischen Problemen; dabei ist die Verwendung des Begriffs „Softwarezuverlässigkeit“ umstritten: der zufällige Charakter hängt in der Tat zusammen mit dem Auftreten der Programmierfehler („Bugs“) und nicht mit den Programmierfehlern selbst, deren Anwesenheit im Code **deterministisch** ist [Bied, 1998]. Ferner hängt der vermeintliche zufallsbedingte Charakter der Mechanismen zum Aufdecken von Programmierfehlern eher mit der Unkenntnis der Entwerfer und Anwender in Bezug auf den Einfluss des Umfeldes auf das Verhalten der Software zusammen.

**Im Folgenden wird aufgezeigt, dass die Einführung von formalen Verfahren stark mit Fachkenntnissen aus dem Bereich der Informatik zusammenhängt. Formale Verfahren beruhen nicht auf der Analogie Hardware-Software, sie betrachten die Software als eigenständige, abstrakte Einheit, die voll und ganz dem menschlichen Geist entsprungen ist und so Gegenstand einer mathematischen Betrachtung werden kann und bei den Eigenschaften bzw. Verhaltensinvarianten, Spezifikationskonformität oder Unabhängigkeit von bestimmten umweltbedingten Zufällen bewiesen werden können.**

**Die praktische Anwendung von formalen Verfahren auf kritische Software von Sicherungsmodulen könnte so zu einem Ansatz der „inhärenten Betriebssicherheit“ oder zu einem „fail safe“**[37]**-Ansatz führen** [Bied, 1998] [Bied, 2003]

[36] Cette notion sera plus amplement définie dans le chapitre 4.

[37] Dieser Begriff wird ausführlicher in Kapitel 4 definiert.

## 5.2.2 Diversité des méthodes formelles

La diversité des méthodes formelles et semi formelles amène naturellement l'utilisateur potentiel à adopter des critères de classification ou de regroupement. Différents points de vue peuvent être appliqués et il est intéressant de les évoquer successivement afin de justifier notre choix final.

### 5.2.2.1 Le point de vue des langages

L'application d'une méthode formelle à un logiciel passe nécessairement par une mathématisation du contenu de ce logiciel. Cette mathématisation, même si elle peut être très complète et couvrir un périmètre très large rendant compte de son comportement, est forcément partielle, car elle est liée à des postulats de base, à un référentiel d'hypothèses concernant l'ensemble des données d'entrées conditionnant le comportement de ce logiciel : domaine de validité, plages de variation, dynamique d'évolution, etc.

Dans le cadre des méthodes formelles, cette mathématisation du contenu du logiciel repose sur l'utilisation de modèles eux-mêmes basés sur la mise en œuvre de langages de modélisation.

Ces modèles, bien que nécessitant souvent des compétences spécialisées concernant par exemple la connaissance de la grammaire ou de la syntaxe des langages associés, présentent de nombreux avantages : l'avantage le plus important est de pouvoir spécifier différents niveaux d'abstraction susceptibles de faciliter la gestion de la complexité inhérente aux applications.

Des modèles très abstraits sont par exemple utilisés pour présenter l'architecture générale d'une application ou sa place dans une organisation, tandis que des modèles très concrets permettent de spécifier très précisément des protocoles de communication réseau ou des algorithmes de synchronisation. Même si des modèles peuvent se situer à des niveaux d'abstraction très différents, il est possible de définir des relations de raffinement entre eux. Véritables liens de traçabilité, ces relations sont garantes de la cohérence d'un ensemble de modèles représentant une même application.

## 5.2.2 Vielfalt formaler Methoden

Die Vielfalt formaler und semiformaler Methoden veranlasst den potenziellen Benutzer, Klassifizierungs- oder Gruppierungskriterien einzuführen. Dies kann unter unterschiedlichen Gesichtspunkten geschehen und es ist interessant, diese im Folgenden darzustellen, um die endgültige Wahl zu rechtfertigen.

### 5.2.2.1 Abstraktionsniveau

Die Anwendung einer formalen Methode geht notwendigerweise mit einer mathematischen Formulierung des Softwareinhalts einher. Diese mathematische Formulierung ist, selbst wenn sie sehr vollständig ist und einen sehr breiten Bereich abdeckt, notgedrungenerweise partiell. Sie hängt von den Grundanforderungen ab und von allen Hypothesen bezüglich der Eingangsbereiche, die das Verhalten der Software beeinflussen: Gültigkeitsbereich, Variationsbereiche, Entwicklungsdynamik usw.

Im Rahmen der formalen Methoden beruht diese „mathematische Formulierung“ des Softwareinhalts auf der Benutzung von Modellen, die selbst auf Modellsprachen basieren.

Obwohl diese Modelle oft spezielle Fachkenntnisse erfordern (zum Beispiel die Kenntnis der Grammatik oder der Syntax der assoziierten Sprachen), haben sie zahlreiche Vorteile: der Wichtigste besteht darin, verschiedene Abstraktionsniveaus spezifizieren zu können, welche die Verwaltung der Komplexität der Anwendung vereinfachen können.

Sehr abstrakte Modelle werden benutzt, um die allgemeine Architektur einer Anwendung oder ihre Stellung in einer Organisation darzustellen. Sehr konkrete Modelle erlauben es, Netzkommunikationsprotokolle oder Synchronisierungsalgorithmen genau zu spezifizieren. Selbst wenn Modelle unterschiedliche Abstraktionsniveaus haben können, ist es möglich, Vefeinerungsbeziehungen untereinander zu definieren. Diese Verbindungen erlauben eine Rückverfolgung und garantieren so die Kohärenz aller Modelle derselben Anwendung.

La diversité des possibilités de modélisation, ainsi que la possibilité d'exprimer des liens de traçabilité constituent des atouts décisifs pour gérer la complexité. Cette richesse est liée à la diversité des langages et à leurs propriétés ou caractéristiques, qui renvoient à des notions:

- purement mathématiques: langages dits algébriques ;
- comportementales : langages asynchrones ;
- graphiques, topologiques ou structurels, en relation avec des formalismes de représentation graphique : langages à flux communicant.

On peut citer à titre d'information la logique de classification rencontrée dans la littérature [IMdR, 2009] :

- Modèles à flux communicant : réseaux de Petri, Statecharts, SDL, GRAFCET…
- Modèles synchrones : ESTEREL, LUSTRE, Signal…
- Modèles algébriques : langages B, Z, VDM…

Ces langages formels peuvent aussi être abordés sous d'autres points de vue.

### 5.2.2.2 Le point de vue des traitements

Les méthodes formelles peuvent être classifiées suivant les traitements mathématiques qui vont être appliqués aux modèles réalisés à partir de ces langages. En effet, différents corpus mathématiques ont été utilisés pour élaborer des raisonnements formels sur les logiciels et c'est cette même diversité d'approche qui a pu engendrer des «familles» de méthodes formelles.

Afin d'illustrer plus concrètement cette logique de classification, on peut évoquer cette typologie de traitement ou résolution mathématique souvent mise en œuvre dans le cadre de l'exploitation des méthodes formelles :

- le Model Checking effectue une exploration exhaustive de l'évolution du système lors de ses exécutions possibles. Par exemple, pour démontrer l'absence d'erreurs à l'exécution, on teste l'absence d'états d'erreur dans l'ensemble des états accessibles du système. Il s'agit alors d'une forme de test (informatique) exhaustif, mais mené à l'aide d'algorithmes astucieux permettant d'énumérer les états du système sous une forme symbolique économique. Ces tests peuvent être conduits par interprétation directe des GRAFCET ou réseaux de Petri constituant ces modèles,

Die Vielfalt der Modellierungsmöglichkeiten und die Möglichkeit, Rückverfolgbarkeitsverbindungen auszudrücken, sind entscheidende Vorteile beim Umgang mit der Komplexität. Dieser Reichtum beruht auf der Vielfalt der Sprachen und ihren Merkmalen und Eigenschaften. Diese verschiedenen Konzepte sind:

- mathematisch: so genannte algebraische Sprachen
- verhaltensspezifisch: asynchrone Sprachen
- Sprachen mit kommunizierenden Flüssen
- graphisch, topographisch oder strukturell, in Verbindung mit einer Formalisierung der graphischen Darstellung.

Die in der Literatur [IMdR, 2009] benützte Klassifizierung unterscheidet:

- Modelle mit kommunizierenden Flüssen: Petrinetze, Statecharts, SDL, GRAFCET
- Synchrone Modelle: ESTEREL, Signal
- Algebraische Modelle: B-Sprachen, Z, VDM…

Diese formalen Sprachen können auch unter anderen Gesichtspunkten behandelt werden.

### 5.2.2.2 Anwendungsgesichtspunkte

Formale Methoden können nach den mathematischen Methoden klassifiziert werden, die auf die mit Hilfe der Sprachen erstellten Modelle angewendet werden. Verschiedene mathematische Theorien werden benützt, um formale Überlegungen auf Software auszudehnen. Diese Konzeptvielfalt hat „Familien" von formalen Methoden entstehen lassen.

Um diese Logik konkret zu illustrieren, kann man die unterschiedlichen Methoden oder mathematischen Lösungsansätze nennen, die oft bei der Anwendung formaler Methoden eingesetzt werden:

- Das Model checking führt eine komplette Untersuchung der Systementwicklung unter allen Benützungsmöglichkeiten durch. Um das Fehlen von Fehlern bei der Ausführung zu beweisen, testet man das Fehlen von Fehlerzuständen in allen erreichbaren Zuständen des Systems. Es handelt sich dann um eine Art erschöpfenden (Informatik-)Test. Mittels geschickter Algorithmen durchgeführt erlaubt das model checking, die Zustände des Systems in einer symbolischen Form aufzuzählen. Diese Tests können durch direkte Interpretation der GRAFCET oder Petrinetze durchgeführt werden aus denen diese Modelle bestehen.

- une variante du model checking consiste à ne pas analyser directement le système, mais à analyser plutôt un modèle informatique, plus ou moins abstrait par rapport à la réalité ;
- l'analyse statique par interprétation abstraite, schématiquement, calcule symboliquement l'ensemble des états accessibles du système ;
- la preuve automatique de théorème tend à démontrer automatiquement des théorèmes sur les formules logiques décrivant la sémantique du programme ;
- les *assistants de preuve* permettent à l'utilisateur de démontrer des théorèmes sur le programme, avec une aide (plus ou moins grande) et une vérification par la machine.

Les méthodes formelles peuvent s'appliquer à différents stades du processus de développement d'un système (logiciel, électronique, mixte), de la spécification jusqu'à la réalisation finale.

### 5.2.2.3 Le point de vue de la sémantique

Le point de vue des sémantiques et des niveaux d'abstraction peut également être exploité pour classifier les méthodes formelles. Ainsi que vu précédemment, tout modèle mis en œuvre dans le cadre d'une méthode formelle repose sur l'utilisation d'un langage : tout langage peut alors être traité en deux étapes : une analyse syntaxique et un calcul sémantique.

L'analyse syntaxique a pour objectif de transformer une description par suite de caractères (langage "textuel") en une structure appelée arbre de syntaxe abstraite, en conformité avec un ensemble de règles de dérivation décrit très souvent par une grammaire BNF[38] (cas pour les langages textuels, les langages graphiques permettant de construire une description "directe" d'une telle structure). Cet arbre de syntaxe abstraite est l'entrée du calcul sémantique qui se trouve être une fonction récursive sur les nœuds de l'arbre. A chaque nœud une évaluation est réalisée en fonction de la nature du nœud et de l'évaluation (récursive) des branches de ce nœud. [IMdR, 2009]

Plus une sémantique est abstraite, plus celle-ci se rapproche de ce qu'on a l'habitude de qualifier de formel [IMdR, 2009].

Ainsi, le résultat d'une sémantique abstraite est une forme d'évaluation de propriétés à caractère exhaustif, ce qui est ce sur quoi on s'entend lorsqu'on parle de formel.

- Eine Variante des model checking besteht darin, das System nicht direkt zu analysieren, sondern ein mehr oder weniger von der Wirklichkeit abstrahiertes Informatikmodell.
- Die statische Analyse errechnet durch abstrakte Interpretation symbolisch alle erreichbaren Zustände des Systems.
- Der automatische Beweis versucht Annahmen über die logischen Formeln, die die Semantik des Programms beschreiben, zu beweisen.
- Beweisassistenten erlauben es dem Benutzer, Annahmen über das Programm mit einer (mehr oder weniger großen) Hilfe und einer automatischen Überprüfung zu beweisen.

Formale Methoden können auf verschiedene Etappen des Entwicklungsvorgangs eines Systems (Software, Elektronik, Mischung der beiden) angewandt werden, von der Spezifizierung bis hin zur endgültigen Umsetzung.

### 5.2.2.3 Syntax und Semantik

Der Gesichtspunkt der Semantik und des Abstraktionsniveaus kann ebenfalls dazu benützt werden, die formalen Methoden zu klassifizieren. Jedes im Rahmen einer formalen Methode verwirklichte Modell beruht auf der Benutzung einer Sprache. Jede Sprache kann in zwei Etappen bearbeitet werden: die Analyse der Syntax und die semantische Berechnung.
Die syntaktische Analyse hat zum Ziel, eine Beschreibung aus Buchstabenfolgen („textuelle" Sprache) in einen sogenannten Baum abstrakter Syntax umzuwandeln. Dies geschieht in Übereinstimmung mit Ableitungsregeln, die sehr oft durch eine Grammatik beschrieben werden (dies ist der Fall bei textuellen Sprachen; graphische Sprachen erlauben die „direkte" Beschreibung einer solchen Struktur). Dieser Baum abstrakter Syntax ist die Eingabegröße für die semantische Berechnung, die rekursiv an jedem Knoten des Baums arbeitet. An jedem Knoten wird eine Berechnung durchgeführt in Funktion der Art des Knotens und der Bewertung der Kanten dieses Knotens. [IMdR, 2009]

Je abstrakter eine Semantik ist, desto mehr nähert sie sich dem an, was man als formal bezeichnet [IMdR, 2009].

Das Ergebnis einer abstrakten Semantik ist eine Art der erschöpfenden Bewertung der Eigenschaften. Dies ist es, was man unter „formal" versteht.

[38] Bibliothèque Nationale de France

### 5.2.2.4 Les catégories de méthodes formelles

Génération automatique de codes

**La génération automatique de codes exécutables** transforme une description, dans un langage de très haut niveau et très spécialisé, en un programme dans un langage classique (C ou ADA). Les preuves formelles établies sur le modèle sous-tendant la description abstraite sont applicables au programme source constitué par le code généré automatiquement. Cette technique est bien adaptée pour les systèmes temps réels.

Vérificateurs de modèles (model checkers)

**Les vérificateurs de modèles** sont des outils qui permettent de décider si un modèle d'un système satisfait ou non des propriétés logiques. Dans cette démarche, le modèle est une représentation du système, et les propriétés représentent les fonctionnalités attendues du système. Cette technique est très bien adaptée pour la conception des systèmes «temps – réel», mais manipule assez mal les données. En pratique, les utilisateurs appliquent manuellement des techniques d'interprétation abstraite lorsque le système possède de trop nombreuses données.

### 5.2.2.4 Kategorisierung formaler Methoden

Automatische Codegenerierung

**Die automatische Generierung ausführbarer Codes** wandelt eine Beschreibung in einer sehr speziellen Sprache hohen Niveaus in ein Programm klassischer Programmiersprache (C oder ADA) um. Die formalen Beweise, die auf der dem Modell zugrunde liegenden abstrakten Beschreibung beruhen, können auf den Quellcode angewandt werden, der automatisch erzeugt wurde. Diese Technik ist gut für Echtzeitsysteme geeignet.

Modellprüfer (model checkers)

**Modellprüfer** sind Werkzeuge, die es erlauben zu entscheiden, ob das Modell eines Systems die logischen Eigenschaften erfüllt. Bei diesem Vorgehen ist das Modell eine Darstellung des Systems. Eigenschaften stellen die vom System erwarteten Funktionalitäten dar. Diese Technik ist vorteilhaft für die Konzeption von Echtzeitsystemen, handhabt Daten jedoch ziemlich schlecht. In der Praxis wenden die Benutzer manuell die Techniken abstrakter Interpretation an, was zur Folge hat, dass das System zu viele Daten besitzt.

## 5.2.3 Conception validation formelle

La conception ou la modification d'un système informatique critique requiert une validation de ses fonctionnalités par des méthodes de preuve formelle afin de **garantir la correction de ses fonctionnalités**. Il s'agit d'une question cruciale aujourd'hui pour l'ensemble du secteur informatique aéronautique [Leveson, 2001] ferroviaire et automobile : comment spécifier et réaliser correctement un logiciel, un circuit etc. Le coût de validation, de test et de mise au point des logiciels est alors très supérieur à celui de leur développement.

Ce problème est particulièrement difficile, et même indécidable dans le cas général si des précautions n'ont pas été prises lors de la conception. Deux grandes voies sont actuellement explorées :

- la première consiste à prouver la conformité entre une réalisation et une spécification abstraite supposée correcte ;
- la seconde cherche à prouver la spécification elle-même.

Ces deux approches sont complémentaires.

**NB** : Il convient de préciser le sens donné aux mots « vérification » et « validation » formelles dans ce document :

- vérification [39]
  Il s'agit de répondre à la question : avons-nous fait ce que nous devions faire ?, i.e. le produit est-il conforme aux spécifications ?
- validation [40]
  Il s'agit de répondre à la question : ce que nous faisons est-il conforme aux attentes de la situation ? i.e. le produit va-t-il satisfaire les besoins réels du client ?

## 5.2.3 Formale Entwicklung bzw. Verifikation

Die Entwicklung oder Änderung eines kritischen IT-Systems erfordert zur **Gewährleistung der Korrektheit der Funktionalitäten** die Validierung der Funktionen durch formale Beweisführungsverfahren. Dies ist heute eine kritische Frage für den IT-Bereich sowohl im Luftfahrt- [Leveson, 2001], Eisenbahn- oder Automobilsektor. Wie kann man eine Software, eine Schaltung, usw. korrekt spezifizieren und realisieren? Die Kosten der Validierung, der Tests und der Feineinstellung der Software sind sehr viel höher als deren Entwicklungskosten.
Dieses Problem ist besonders schwierig, im Allgemeinen sogar ohne Lösung, falls bei der Softwareentwicklung nicht schon Vorsichtsmaßnahmen getroffen wurden. Derzeit werden zwei Möglichkeiten erforscht:

- Die erste prüft die Konformität zwischen der Durchführung und der abstrakten Spezifizierung, die als korrekt angenommen wird.
- Die zweite versucht, die Spezifikation selbst zu testen.

Beide Ansätze ergänzen einander.

**NB**: Es empfiehlt sich den Sinn der in diesem Dokument angegebenen Wörter „Verifikation" und „Validierung" zu präzisieren:

- Verifikation [39]
  Es handelt sich darum, auf die Frage zu antworten: Haben wir gemacht, was wir machen sollen? I.e. ist das Produkt spezifikationsgemäß?
- Validation [40]
  Es handelt sich darum, auf die Frage zu antworten: Was machen wir, sind die Erwartungen der Situation gemäß? I.e. wird das Produkt die wirklichen Bedürfnisse des Kunden zufrieden stellen?

[39] Verification: have we made what we were trying to make? i.e. does the product conform to the specification?

[40] Validation : are wie trying to make the right thing ?, i;e. does the product do what the user really requiers?

### 5.2.3.1 Preuve de conformité

C'est le domaine où l'on a le plus de résultats. Trois grandes idées peuvent être mises en œuvre :
- considérer le système comme un automate à états finis dont on explore les états accessibles ;
- utiliser des « prouveurs » automatiques de théorèmes (directement sur du code particulier éventuellement issu d'une transformation d'automatismes),
- synthétiser directement la réalisation à partir d'une syntaxe rigoureuse (écriture directe avec les invariants…).

Exploration d'états

Tous les systèmes effectivement réalisés, dans la mesure où ils fonctionnent sur des ordinateurs dont la mémoire est limitée, peuvent être considérés (sans erreur matériel ou interruptions non maîtrisées) comme des **automates à états finis**.

Partant de ce constat, on peut construire le produit d'un automate qui représente la spécification par un deuxième automate qui représente sa réalisation, les sorties des deux automates étant réunies dans un comparateur (XOR). Pour prouver que la réalisation est conforme à la spécification, il suffit alors de montrer que les sorties de l'automate produit sont toujours nulles. On est donc ramené au problème d'explorer l'ensemble des états accessibles d'un automate à états finis. Il existe aujourd'hui des techniques efficaces pour cela par exemple les arbres binaires BDD (Binary Decision Diagram).

Cette méthode donne d'excellents résultats dans le domaine des circuits digitaux synchrones [Bielinski, 1993].

**Cette méthode pose toutefois deux difficultés (pour leur application sur les logiciels temps réel) :**
- **La première est qu'elle se généralise difficilement au cas du logiciel, qui est rarement conçue comme un automate ;**
- **La seconde, plus gênante, est qu'un automate ne fonctionne jamais seul : on lui fournit généralement des informations sous la forme de données ou d'instructions (code…). Il faut donc ajouter aux états de l'automate ceux de sa mémoire ou de son environnement. Cela conduit à une explosion combinatoire. De plus, la validation obtenue dépend des valeurs initiales fixées pour la mémoire. Elle est donc plus complète qu'un simple test, mais loin d'être exhaustive.**

### 5.2.3.1 Konformitätsbeweis

Dies ist das Gebiet, auf dem es die meisten Ergebnisse gibt. Drei Ideen können umgesetzt werden:
- das System als endlichen Automaten zu betrachten bei dem man die zugänglichen Zustände untersucht
- Beweisautomaten zu verwenden (direkt auf dem speziellen Code, der eventuell durch eine automatische Umwandlung entstanden ist)
- eine Programmsynthese von einer präzisen Syntax ausgehend durchzuführen (Schreiben mit Hilfe von Invarianten…).

Zustandsuntersuchung

Alle tatsächlich verwirklichten Systeme, so weit sie auf Rechnern laufen, deren Arbeitsspeicher begrenzt ist, können (ohne Materialfehler oder nicht beherrschte Unterbrechungen) **als endliche Automaten betrachtet werden**.

Von dieser Feststellung ausgehend, kann man ein Ergebnis erstellen bei dem die Spezifikationen durch einen ersten Automaten dargestellt werden und die Umsetzung durch einen zweiten. Die Ausgänge der beiden werden durch einen Vergleicher (XOR) verbunden. Um zu beweisen, dass die Umsetzung mit der Spezifizierung im Einklang steht, reicht es dann aus zu zeigen, dass die Ausgänge des Produktautomaten immer gleich null sind. Man kommt also auf das Problem zurück, alle zugänglichen Zustände eines endlichen Automaten zu untersuchen. Hierfür gibt es heute effiziente Techniken, zum Beispiel binäre Bäume (BDD).

Diese Methode führt zu hervorragenden Ergebnissen im Bereich der synchronen digitalen Kreisläufe [Bielinski, 1993].

**Diese Methode beinhaltet allerdings zwei Schwierigkeiten (bei ihre Anwendung auf Echtzeitsoftware):**
- **Die erste ist, dass sie sich schwer auf Software verallgemeinern lässt, die selten als ein Automat entworfen wird.**
- **Die zweite ist, dass ein Automat niemals alleine funktioniert: man liefert ihm im Allgemeinen Informationen in Form von Daten oder Befehlen (Code…). Man muss also zu den Zuständen des Automaten jene seines Gedächtnisses oder seiner Umwelt hinzufügen. Das führt zu einer kombinatorischen Explosion. Außerdem hängt die erhaltene Validierung von den Anfangswerten ab, die für den Speicher festgelegt wurden. Die Methode ist also vollständiger als ein einfacher Test, aber noch nicht erschöpfend.**

Utilisation de prouveurs

Une seconde approche consiste à prouver mathématiquement l'équivalence, en un certains sens, entre la spécification et sa réalisation. Étant données la longueur et la complexité de ce travail, l'utilisation de prouveurs automatiques est alors indispensable. Malheureusement il est indispensable de leur enseigner les mathématiques, sous la forme de longues bibliothèques de théorèmes qui concernent à la fois le formalisme des objets manipulés par le système à valider (addition, affectations, listes…), le formalisme de description du système lui-même (octets, variables…) et la notion d'équivalence entre une spécification et sa réalisation. De plus, les prouveurs de théorème ne sont capables de prouver directement, un à un, que des énoncés simples. Il est nécessaire de leur indiquer de nombreux points de passage de la démonstration. Les validations ainsi obtenues sont donc exhaustives mais longues et demandent une participation importante de l'utilisateur.

Il existe des prouveurs universitaires avec des bibliothèques de théorèmes [Gardey, 2005]. Ces prouveurs requièrent de l'utilisateur des compétences mathématiques spécifiques. Ces résultats sont de nature universitaire et rarement industrielle [Bitsch, 2000] [Bitsch, 2002] [Bitsch, 2003].

**L'écriture des propriétés à démontrer en un langage « prouvable » constitue, pour ma part, une réelle difficulté qui pousse les industriels à abandonner cette voie.**

Conception formelle

La troisième technique consiste à synthétiser directement la réalisation à partir de la spécification (et non l'expression réelle des besoins système), ce qui suppose d'écrire cette dernière dans un langage dont la sémantique est parfaitement définie (langage B par exemple). Cette approche est surtout utilisée pour les systèmes tels que les automates, contrôleurs, interface homme machine et autres applications temps réel. C'est l'approche, essentiellement française, de la « programmation synchrone ».

Ainsi ETEREL est un langage abstrait de spécification d'automates. Il permet de manipuler des automates ayant un nombre d'états trop élevé pour être explicités. La traduction en automates peut être prouvée automatiquement dans certains cas. ESTEREL est efficace pour aborder les problèmes dont le séquencement est complexe.

Lorsque la complexité provient, non du séquencement, mais des opérations de base, on peut utiliser d'autres langages synchrones comme LUSTRE.

Benutzung von automatischen Beweisen

Ein zweites Konzept besteht darin, die Gleichwertigkeit zwischen der Spezifizierung und ihrer Umsetzung mathematisch zu beweisen. Aufgrund der Dauer und der Komplexität dieser Aufgabe ist die Benutzung von automatischen Beweisen unentbehrlich. Leider ist es auch unentbehrlich, den Beweisen die Mathematik, in Form von Theorembibliotheken, „beizubringen". Diese betreffen gleichzeitig den Formalismus der Objekte, die von dem zu überprüfenden System gehandhabt wurden (Addierung, Zuordnungen, Listen…), den Formalismus der Beschreibung des Systems selbst (Bytes, Variablen…) und das Gleichwertigkeitskonzept zwischen einer Spezifizierung und ihrer Umsetzung. Die Beweiser können nur einfache Eigenschaften, eine nach der anderen, direkt beweisen. Es ist notwendig, ihnen zahlreiche Etappenziele des Beweises anzugeben. Die so erhaltenen Validierungen sind also erschöpfend, dauern aber lange und verlangen eine große Beteiligung des Benutzers.

Es gibt Hochschulbeweiser mit Theorembibliotheken [Gardey, 2005]. Diese Beweisautomaten erfordern eine spezifische mathematische Kompetenz des Benutzten. Die Ergebnisse dieser Methode sind akademisch und meistens nicht industriell [Bitsch, 2000] [Bitsch, 2002] [Bitsch, 2003]. **Das Schreiben der zu beweisenden Eigenschaften in einer beweisbaren Sprache stellt für mich eine wirkliche Schwierigkeit dar, die die Unternehmen dazu bringt, diesen Weg aufzugeben.**

Formale Konzeption

Die dritte Technik besteht in der direkten Synthese, der Spezifizierung in der Realisierung (keine Formulierung der wirklichen Anforderungen an das System). Dies erfordert, die Spezifikationen in einer Sprache zu schreiben, deren Semantik vollkommen definiert ist (B-Sprache zum Beispiel). Dieses Konzept wird besonders bei Systemen wie Automaten, Controller, Mensch-Maschineschnittstelle und anderen Echtzeitanwendungen benutzt. Dies ist das, vor allem französische, Konzept der „synchronen Programmierung".

ETEREL ist eine abstrakte Sprache der Automatenspezifizierung. Sie erlaubt es, Automaten zu handhaben, die eine zu hohe Zustandsanzahl haben, um explizit beschrieben zu werden. Die Übersetzung in Automaten kann in bestimmten Fällen automatisch bewiesen werden. ESTEREL ist gut dafür geeignet, um Probleme anzugehen, deren Abfolge komplex ist.

Wenn die Komplexität nicht von der Abfolge sondern von den Basisoperationen stammt, kann man andere synchrone Sprachen wie LUSTRE benutzen.

Les spécifications sont alors écrites sous la forme d'équations reliant les variables de l'instant *t* à celles de l'instants *t*+1. Ces outils donnent de bons résultats dans le domaine des circuits réactifs. Ils sont en cours d'industrialisation, en particulier chez Dassault et EDF en France.

**Problème résiduel : les langages interprétables par les ordinateurs ne sont généralement pas prouvables, les langages prouvables ne sont généralement pas interprétables par les ordinateurs, il y a donc nécessité d'une transformation d'un langage formel à un langage qui ne l'est pas, ce qui remet en cause la validité de la preuve effectuée au niveau d'abstraction supérieur.**

### 5.2.3.2 Preuve de spécifications

Prouver la conformité d'une réalisation (à partir d'une spécification fournie par la MOA et comprises par une MOE) ne suffit pas pour s'assurer de sa correction : encore faut-il que la spécification dont elle découle ait un sens (dans les modes normaux et dégradés de l'exploitation du système).

Une autre voie explorée (preuve de propriétés) par les méthodes formelles concerne donc directement les spécifications. Peu de résultats sont disponibles, les expressions « preuve formelle » ou « preuve de propriétés », concernent rarement le problème de la validité des spécifications mais les prennent comme données d'entrée.

Or nous savons que les accidents survenus ces dernières années sur des systèmes informatisés sûrs relèvent bien d'avantage des erreurs de spécification [Gartner, 2009] (commune sur les *N* unités en parallèle) que des erreurs de réalisation (codage ou matériel) activées par une combinaison particulière des entrées (même interprétation sur les *N* unités en parallèle). [Nguyen, 2008] [Lötschberg, 2008] [SNCF, 2005] [Bombardier, 2005]

Les questions qui se posent sont « la spécification est-elle correcte ? » et « qu'est-ce qu'une spécification correcte ? ». Quatre grandes approches peuvent être envisagées :

- une limitation à des cas particuliers ;
- la méthode des invariants ;
- l'utilisation d'un prouveur ;
- l'exécution directe d'une preuve

In diesem Fall werden die Spezifikationen in Form von Gleichungen beschrieben, die die Variablen zum Zeitpunkt *t* mit den Variablen zum Zeitpunkt *t*+1 verbinden. Diese Werkzeuge ergeben gute Ergebnisse im Bereich der reaktiven Schaltungen. Sie werden derzeit industrialisiert, insbesondere bei Dassault und EDF in Frankreich.

**Ein Problem bleibt bestehen: die von Computern interpretierbaren Sprachen lassen sich im Allgemeinen nicht beweisen. Beweisbare Sprachen lassen sich im Allgemeinen nicht von Computern überprüfen. Deshalb müssen formale Sprachen in nicht formale Sprachen umgewandelt werden. Dadurch wird jedoch die Gültigkeit des Beweises der übergeordneten Abstraktionsebene in Frage gestellt.**

### 5.2.3.2 Spezifizierungsbeweis

Der Konformitätsbeweis einer Umsetzung (aufgrund von einer vom Auftraggeber gelieferten Spezifikation, die vom Auftragnehmer übernommen wird) reicht nicht aus, um deren Korrektheit zu beweisen: dazu muss die Spezifizierung, von der sie abgeleitet wurde, auch einen Sinn machen (in der normalen Funktionsweise des Systems und in der Rückfallebene).

Eine weitere Anwendung (Eigenschaftsbeweis) formaler Verfahren betrifft also direkt diese Spezifikationen. Hierzu sind wenige wissenschaftliche Ergebnisse verfügbar, die Begriffe „formaler Beweis" oder „Eigenschaftsbeweis" betreffen selten die Gültigkeit der Spezifikationen; diese werden jedoch als Eingangsinformationen verwendet.

Nun ist aber bekannt, dass die Unfälle der letzten Jahre (in Verbindung mit sicheren IT-Systemen) viel mehr auf Spezifizierungsfehler (gemeinsame Interpretation auf *N* parallelgeschalteten Einheiten) [Gartner, 2009] als auf von einer besonderen Eingangskombination (dieselbe Interpretation auf den *N* parallelgeschalteten Einheiten) hervorgerufene Umsetzungsfehler (Kodifizierung oder Hardware) zurückzuführen sind. [Nguyen, 2008] [Lötschberg, 2008] [SNCF, 2005] [Bombardier, 2005]

Es stellen sich also folgende Fragen: „Ist die Spezifikation korrekt?" bzw. „Was ist eine korrekte Spezifikation?". Vier Hauptansätze können dafür in Erwägung gezogen werden:

- Beschränkung auf Sonderfälle
- Verfahren der Invarianten
- Verwendung eines Beweisautomaten
- direkte Ausführung einer Beweisführung

Limitation à des cas particuliers

Il s'agit de vérifier des propriétés décidables : existence de «dead locks» ou propriétés logiques simples du type « au bout de trois cycles telle variable contient telle valeur ». Cette approche est toutefois très limitée et ne peut servir dans le cadre d'une preuve exhaustive.

Méthode des invariants

La méthode des invariants est relativement ancienne [Narboni, 2001]. Elle s'applique d'une manière générale à des algorithmes et consiste à associer à chaque étape d'un calcul une propriété logique, un invariant) qui relie les variables entre elles. L'un de ces invariants décrit le résultat obtenu à la fin du calcul, ce qui valide l'algorithme.

Bien que très puissante, cette méthode est lourde à mettre en œuvre sur le plan industriel, ce qui a conduit à un relatif abandon. En effet, les invariants doivent être élaborés intelligemment (manuellement) et font partie intégrante du processus de spécification : il est très difficile de définir des invariants d'une spécification qui n'a pas été écrite dans ce but. Les langages Z et B en donnent une bonne illustration. Des ateliers spécialisés permettent d'assister le concepteur tout en requérant de fortes compétences mathématiques.

**Cette approche n'est applicable qu'à la conception ex nihilo, ce qui est rarement le cas dans le monde industriel et explique sa très faible application industrielle**.

Utilisation d'un prouveur

Les prouveurs peuvent être utilisés, pour peu que l'on sache traduire sous forme d'un théorème à démontrer la notion de correction d'une spécification. C'est le cas particulier des systèmes dont l'objectif est de calculer une fonction mathématique bien définie. Par exemple, étant donnés deux entiers *A*, *B* et le résultat calculé *C*, il faut vérifier que $C = A \times B$ (ou calcul d'un code CRC…).

Exécution directe d'une preuve

Il s'agit d'une idée développée par l'INRIA. Il s'agit d'extraire automatiquement un programme à partir de la preuve constructive d'existence d'une solution à un problème donné. Le formalisme retenu est celui de la programmation fonctionnelle avec le langage CAML.

**Là encore on est majoritairement dans le monde universitaire et non industriel.**

Beschränkung auf Sonderfälle

Hier geht es darum, entscheidbare Eigenschaften zu prüfen: Bestehen von „dead locks" oder einfache logische Eigenschaften (z.B. „nach drei Zyklen hat diese oder jene Variable diesen oder jenen Wert"). Dieser Ansatz ist jedoch sehr beschränkt und ermöglicht keine vollständige Beweisführung.

Verfahren der Invarianten

Das Verfahren der Invarianten ist relativ alt [Narboni, 2001]. Es ist im Allgemeinen auf Algorithmen anwendbar. Mit jeder Etappe einer Berechnung wird eine logische Eigenschaft assoziiert (eine Invariante); diese verbindet die Variablen untereinander. Eine dieser Invarianten beschreibt das Ergebnis am Ende der Berechnung, was den Algorithmus validiert.

Obwohl es sehr leistungsfähig ist, ist dieses Verfahren auf industrieller Ebene schwer zu implementieren, weswegen es so gut wie aufgegeben wurde. Die Invarianten müssen in der Tat auf intelligente Weise (manuell) erstellt werden, sie sind ins Spezifizierungsverfahren integriert: es ist sehr schwer, die Invarianten einer Spezifizierung, die nicht zu diesem Zweck geschrieben wurde, zu definieren. Spezialisierte Arbeitsgruppen unterstützen den Entwickler, erfordern jedoch solide mathematische Kompetenzen.

**Dieser Ansatz ist nur auf ex nihilo Entwicklungen anwendbar. Da dies in der Industrie jedoch selten der Fall ist, wird dieser Ansatz dort selten angewandt.**

Verwendung eines Beweisautomaten

Beweisautomaten können verwendet werden, wenn man den Begriff der Korrektur einer Spezifikation in einen beweisbaren Satz umsetzen kann. Das ist der Fall bei Systemen deren Ziel darin besteht, eine definierte mathematische Funktion zu berechnen, z.B., wenn man zwei ganzzahlige Zahlen *A* und *B* hat, und wenn das Ergebnis der Berechnung *C* ist, dann muss man prüfen, ob $C = A \times B$ (oder Berechnung eines CRC-Code…).

Direkte Ausführung einer Beweisführung

Dabei handelt es sich um eine vom INRIA entwickelte Idee: ein Programm wird automatisch vom konstruktiven Existenzbeweis einer Lösung eines gegebenen Problems abgeleitet. Der gewählte Formalismus ist die funktionelle Programmierung mit der CAML-Sprache.

**Auch hier befindet man sich eher in der akademischen als in der industriellen Welt.**

## 5.2.4 L'utilisation des méthodes formelles

Le développement d'un logiciel nécessite classiquement la réalisation de différents modèles correspondant à différents niveaux d'abstraction, depuis la spécification jusqu'au codage.

### 5.2.4.1 Les différents modèles utilisés dans le cycle de développement

Les méthodes formelles peuvent être utilisées à différentes étapes d'un cycle de développement, pour donner une spécification du système que l'on souhaite développer, au niveau de détail désiré.

**Une spécification formelle du système est alors basée sur un langage formel dont la sémantique est bien définie (contrairement à une spécification en langage naturel qui peut donner lieu à différentes interprétations).**

Cette description formelle du système peut être utilisée comme référence pendant le développement. De plus, elle peut être utilisée pour vérifier (formellement) que la réalisation finale du système (décrite dans un langage informatique dédié) respecte les attentes initiales (notamment en termes de fonctionnalité).

**Le choix d'une méthode formelle dépend, d'une part, de la nature même du système informatique (de sécurité ou non) à développer, et, d'autre part, du positionnement de celui qui doit faire ce choix, Maîtrise d'Ouvrage ou Maîtrise d'œuvre., et par là de l'étape du projet où elle doit s'appliquer.**

### 5.2.4.2 Spécification

Il est indispensable de déterminer les besoins du logiciel pendant cette première phase. Les besoins peuvent se traduire sous plusieurs formes, des spécifications générales, des spécifications fonctionnelles, des spécifications d'interface :

- les spécifications générales sont un ensemble d'objectifs, de contraintes (utilisation de matériels et outils existants) et de généralités qu'il faudra respecter au cours du développement ;
- les spécifications fonctionnelles sont la description des fonctionnalités du logiciel de manière aussi détaillée que nécessaire ;
- les spécifications d'interfaces sont la description des interfaces du logiciel avec le monde extérieur (hommes, autres logiciels, matériels...) de manière aussi détaillée que nécessaire ;

## 5.2.4 Anwendung formaler Methoden

Die Softwareentwicklung erfordert üblicherweise die Erstellung verschiedener Modelle mit unterschiedlichen Abstraktionsniveaus, von der Spezifizierung bis hin zur Codierung.

### 5.2.4.1 Unterschiedliche Entwicklungsmodelle

Formale Methoden können in verschiedenen Etappen eines Entwicklungszyklus für eine Spezifizierung des Systems benutzt werden und zwar mit dem gewünschten Niveau an Details.

**Eine formale Spezifizierung des Systems beruht auf einer formalen Sprache, deren Semantik genau definiert ist (entgegen einer Spezifizierung in natürlicher Sprache, die verschiedene Interpretationen zulassen kann).**

Diese formale Systembeschreibung kann während der Entwicklung als Referenz dienen. Sie kann auch zur (formalen) Überprüfung der Einhaltung der ursprünglichen Anforderungen (insbesondere im Hinblick auf die Funktionalitäten) beim endgültigen Systems (in einer eigenen IT-Sprache geschrieben) dienen.

**Die Wahl eines formalen Verfahrens hängt einerseits von der Art des zu entwickelnden (Sicherheits-/„Nichtsicherheits"-) IT-Systems ab, andererseits von der Stellung der Person, die diese Entscheidung trifft (Auftraggeber, Auftragnehmer) und somit von der Projektetappe, in der sie angewandt wird.**

### 5.2.4.2 Spezifikation

Es ist notwendig, die Softwareanforderungen in dieser ersten Phase zu definieren: dies erfolgt in verschiedenen Formen (allgemeine Spezifikationen, funktionelle Spezifikationen, Schnittstellenspezifikationen):

- Allgemeine Spezifikationen sind eine Reihe von Zielsetzungen, Beschränkungen (Verwendung von bestehender Hardware bzw. bestehenden Tools) und Allgemeinheiten, die im Laufe der Entwicklung einzuhalten sind.
- Funktionelle Spezifikationen beschreiben so detailliert wie nötig die Softwarefunktionalitäten.
- Schnittestellenspezifikationen beschreiben so detailliert wie nötig die Schnittstellen zwischen der Software und der Außenwelt (Menschen, andere Software, Hardware...).

- les spécifications des postulats servent à définir ce que doit faire le logiciel et non comment il est fait. Ceci est décrit dans le document de spécifications des besoins.

**L'expérience montre que c'est à ce stade que sont faites les erreurs ou imprécisions les plus difficiles à mettre en lumière ultérieurement soit lors du développement, soit lors des essais finaux. En effet, comment bâtir des tests relatifs à un domaine non prévu des entrées, avec des fonctionnalités ou des postulats de fonctionnement non pris en compte à la conception ?**

### 5.2.4.3 Conception préliminaire – conception détaillée

Une première étape dans le processus de conception d'un logiciel, à partir des spécifications des besoins, permet de se focaliser sur la définition de l'architecture du logiciel. La phase de conception préliminaire permet d'envisager plusieurs solutions au problème posé et d'en étudier leur faisabilité. Pour chaque solution, les choix effectués sont notés avec leurs raisons de façon à distinguer les contraintes réelles du projet des contraintes déduites trop hâtivement. La solution répondant le mieux aux besoins exprimés est retenue et figée.

Une deuxième étape dans le processus de conception du logiciel permet, à partir du résultat de la conception générale de poursuivre le découpage du logiciel jusqu'à arriver à une description externe de chacune des procédures et des structures de données. Dans le cas de l'utilisation d'un langage modulaire, cette phase consiste à définir précisément les interfaces des modules. Dans le cas de l'utilisation d'un langage orienté objet, cette phase consiste à définir précisément les contenus des objets: attributs et méthodes.

### 5.2.4.4 Codage

Les procédures identifiées lors de la phase précédente sont codées et testées individuellement. Le produit de cette phase est le code source et les résultats des tests unitaires. Dans le cas de l'utilisation d'un langage modulaire, cette phase consiste à coder les corps des modules en respectant leur interface.

La figure suivante illustre ces différents points de vue d'un usage des méthodes formelles [IMdR, 2009] :

- Die Spezifikation der Anforderungen definiert was die Software können muss und nicht wie sie gestaltet sein soll; dies wird in das Bedarfsspezifikationsdokument eingetragen.

**Die Erfahrung zeigt, dass die später am schwersten aufzuzeigenden Fehler/Ungenauigkeiten (bei der Entwicklung oder bei den letzten Erprobungen) in dieser Phase entstehen. Wie soll man in der Tat Tests gestalten für nicht vorgesehene Eingangbereiche oder für bei der Entwicklung nicht berücksichtigte Funktionalitäten oder Anforderungen?**

### 5.2.4.3 Grobentwurf – detaillierter Entwurf

Während der ersten Etappe des Softwareentwurfs konzentriert man sich, von der Bedarfsspezifikation ausgehend, auf die Definition der Softwarearchitektur. In der Grobentwurfsphase können verschiedene Lösungen in Erwägung gezogen werden, und es kann deren Machbarkeit untersucht werden. Für jede Lösung werden die getroffenen Entscheidungen und die Gründe bewertet, um zwischen tatsächlichen Beschränkungen des Projektes und den übereilt angenommen zu unterscheiden. Die Lösung die den Anforderungen am besten entspricht, wird definitiv festgehalten.

In der zweiten Etappe des Softwareentwurfs wird die Software mit Hilfe der Ergebnisse des allgemeinen Entwurfs weiter aufgeschlüsselt bis hin zu einer externen Beschreibung jeder Vorschrift, sowie der Datenstrukturen. Im Falle der Verwendung einer modularen Sprache besteht diese Phase in der präzisen Definition der Modulschnittstellen. Im Falle der Verwendung einer objektorientierten Sprache besteht diese Phase in der präzisen Definition der Objektinhalte: Attribute und Methoden.

### 5.2.4.4 Codierung

Die in der vorherigen Phase identifizierten Verfahren werden codiert und einzeln getestet. Die Ergebnisse dieser Phase sind der Quellcode und die Resultate der Einzeltests. Im Fall der Verwendung einer modularen Sprache besteht diese Phase in der Codierung der Modulkörper unter Einbeziehung der Schnittstellen. Abb. 5.1 illustriert die verschiedenen Verwendungsmöglichkeiten formaler Verfahren [IMdR, 2009].

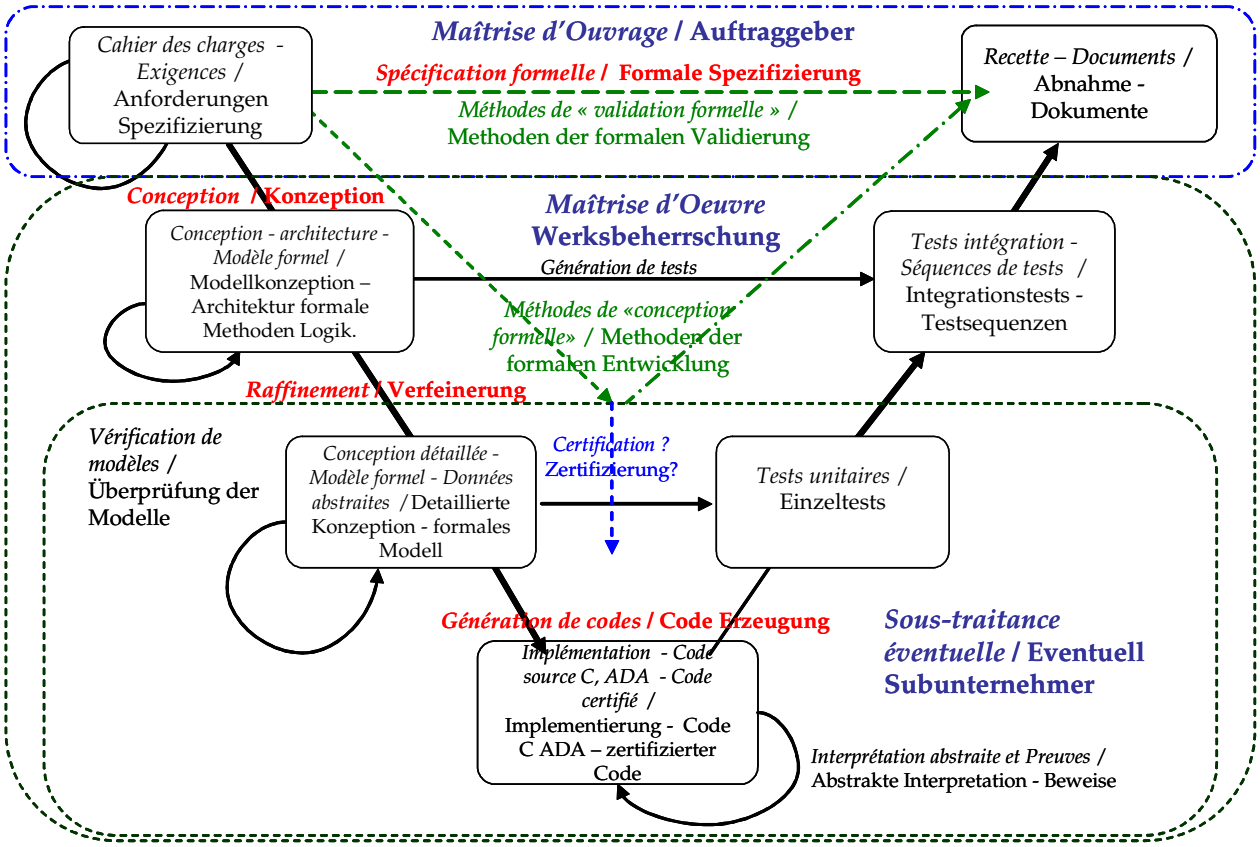



**Figure 5.1 :** Présentation d'un cycle de développement où l'on a introduit la manière dont les méthodes formelles peuvent accompagner le cycle de développement

**Abbildung 5.1:** Darstellung eines Softwareentwicklungszyklus, bei dem gezeigt wird wie formale Methoden den Entwicklungszyklus begleiten

## 5.2.5 Orientations actuelles dans le ferroviaire

Les propriétés des différentes méthodes formelles préjugent de leur utilisation pour tel ou tel système logiciel ou pour tel ou tel niveau d'abstraction :

- Une spécification formelle du système est basée sur un langage formel dont la sémantique est bien définie. Cette description formelle du système peut être utilisée comme référence pendant le développement [Senesi, 2008]. De plus, elle peut être utilisée pour vérifier que la réalisation finale du système respecte les attentes initiales.
- La modélisation formelle du système peut être utilisée comme référence pendant le développement du système concret (mise au point des algorithmes, réalisation en logiciel et/ou circuit électronique).

## 5.2.5 Derzeitige Leitlinien der Eisenbahn

Die Eigenschaften der verschiedenen formalen Verfahren bestimmen deren Verwendung für eine bestimmte Software oder ein bestimmtes Abstraktionsniveau:

- Eine formale Systemspezifizierung beruht auf einer formalen Sprache mit einer genau definierten Semantik. Diese formale Systembeschreibung kann während der Entwicklung als Referenz dienen [Senesi, 2008]. Sie kann auch der Überprüfung der Einhaltung der ursprünglichen Anforderungen bei der endgültigen Realisierung des Systems dienen.
- Die formale Modellierung des Systems kann während der Entwicklung des konkreten Systems als Referenz dienen (Algorithmusfeineinstellung, Umsetzung durch eine Software bzw. eine elektronische Schaltung).

Par exemple :

Si la spécification formelle est dotée d'une sémantique opérationnelle, le comportement observé du système concret peut être comparé avec le comportement de la spécification (qui, elle-même, doit être simulable ou exécutable). De plus, une telle modélisation peut faire l'objet d'une traduction automatique vers le langage cible. Si la spécification formelle est dotée d'une sémantique axiomatique, les *préconditions* et *post conditions* de la spécification peuvent devenir des assertions dans le code exécutable. Il reste à noter que le niveau de « *fiabilité* » de la traduction automatique reste quant à lui très hypothétique notamment quant à son comportement réel sur des machines cibles (hardware et operating system) différentes…

Les méthodes formelles prennent tout leur intérêt lorsque les preuves elles-mêmes sont garanties correctes formellement.

**Lorsque leur pouvoir d'expression le permet, les langages formels sont parfois utilisés comme langage de conception finale. Dans ces cas-là, le code source peut être considéré comme directement écrit dans un langage formel. Ceci n'est pas possible pour tous les langages formels dans la mesure où certains ont été créés afin de décrire des systèmes et non de les réaliser. Cette voie permet de lever les incertitudes liées à la traduction automatique de code et sa compilation au moyen d'un atelier « certifié » (certifié pour un jeu de fonctions et dans un contexte donné ne correspondant pas nécessairement à celui de l'installation à concevoir…).**

Beispiel:

Verfügt die formale Spezifizierung über eine operative Semantik, so kann das beobachtete Verhalten des konkreten Systems mit dem Verhalten der Spezifizierung (die selbst simulierbar oder ausführbar sein muss) verglichen werden. Ferner kann eine solche Modellierung automatisch in die Zielsprache übersetzt werden. Verfügt die Spezifizierung über eine axiomatische Semantik, so können die *Vorbedingungen* und *Nachbedingungen* der Spezifizierung zu Behauptungen im ausführbaren Code werden. Es ist anzumerken, dass der „*Zuverlässigkeitslevel*" der automatischen Umsetzung hypothetisch bleibt, insbesondere im Hinblick auf das tatsächliche Verhalten auf unterschiedlichen Zielmaschinen (Hardware und Betriebssystem).

Die formalen Verfahren gewinnen dann an Interesse, wenn die formale Korrektheit der Beweise selbst gewährleistet ist.

**Wenn die Aussagekraft es erlaubt, werden die formalen Sprachen manchmal als endgültige Entwicklungssprache genutzt. In diesem Fall kann davon ausgegangen werden, dass der Quellcode direkt in einer formalen Sprache geschrieben wurde. Dies ist nicht bei allen formalen Sprachen möglich, zumal einige für die Beschreibung und nicht für die Erstellung bzw. den Entwurf von Systemen entwickelt wurden. Diese Methode erlaubt es, die Unsicherheiten in Verbindung mit der automatischen Codeübersetzung und der Kompilierung mittels (für bestimmte Funktionen und in einem bestimmten Umfeld, das nicht unbedingt dem Umfeld der zu entwickelnden Anlage entspricht) „zertifizierter" Werkzeuge zu beheben.**

### 5.2.5.1 Atelier SCADE

Fondée en 1999, Esterel Technologies [Esterel, 2004] [Esterel, 2008-1] [Esterel, 2008-2] est une filiale de l'INRIA; sa première activité a concerné le développement des logiciels du projet Rafale chez Dassault Aviation. La méthodologie et l'outil sont nés quelques années auparavant et étaient soutenus par Dassault et Thomson-CSF (devenue depuis Thalès). Le produit principal d'Esterel Technologies (SCADE Suite) est un environnement de design et de simulation orienté modèle basé sur UML 2.0, qui génère du code C "sécurisé". L'outil (le code généré) est certifié DO-178B ainsi que IEC61508 et EN50128. [ED12B-DO178B] [DO178B, 1992] [EN50128, 2001]

Les fonctionnalités principales de SCADE sont :

- La représentation visuelle intuitive et familière ;
- La détection précoce des erreurs de design en utilisant des simulations ;
- Code C et Ada généré : lisible, traçable et fiable,
- Tests fonctionnels ;
- Génération automatique de la documentation ;
- Réutilisation des composants.

Un autre produit (SCADE Display) est un outil qui permet de créer les prototypes et de faire la vérification et validation des écrans de visualisation embarqués.

SCADE serait notamment utilisé pour le développement par AnsaldoSTS du Radio Block Centre (RBC) (sans que les essais SNCF ne soient allégés…).

### 5.2.5.1 SCADE

Die 1999 gegründete Fa. Esterel Technologies [Esterel, 2004] [Esterel, 2008-1] [Esterel, 2008-2] ist eine Tochtergesellschaft von INRIA; ihre erste Aufgabe bestand in der Entwicklung der Software des Rafale-Projektes für Dassault Aviation. Die Methodologie und die Werkzeuge entstanden einige Jahre früher, mit der Unterstützung von Dassault und Thomson-CSF (heutige Fa. Thalès). Das Hauptprodukt von Esterel Technologies (SCADE Suite) ist eine modellorientierte Design- und Simulationsumgebung, die auf UML 2.0 beruht und „gesicherten" C-Code generiert. Das Tool (der generierte Code) ist zertifiziert [ED12B-DO178B] [DO178B, 1992] [EN50128, 2001].

Die Hauptfunktionalitäten von SCADE sind:

- eine intuitive und geläufige visuelle Darstellung
- eine frühzeitige Erkennung von Designfehlern mithilfe von Simulationen
- erzeugte C- und Ada-Codes : leserlich, nachvollziehbar und zuverlässig
- funktionelle Tests
- automatische Erstellung der Dokumentation
- Wiederverwendung der Komponenten.

Ein weiteres Produkt (SCADE Display) ist ein Anwendungspogramm zur Erstellung von Prototypen und zur Überprüfung und Validierung von Bordbildschirmen.

Vor allem wurde SCADE anscheinend für die Entwicklung des Radio Block Centre (RBC, Fa. AnsaldoSTS) verwendet (jedoch ohne dass die SNCF-Versuche dadurch erleichtert worden wären).

### 5.2.5.2 Atelier B

La « Méthode B » est une méthode formelle de développement logiciel d'origine française, qui permet de modéliser de façon abstraite dans le langage de B le comportement d'un programme, puis par raffinements successifs, d'aboutir à un modèle concret, sous-ensemble du langage transcodable en Ada ou en C. [IMdR, 2009]

Elle est utilisée pour :

- la réalisation de logiciels corrects par construction ;
- la modélisation de systèmes dans leur environnement.

Le développement de la méthode et des outils est assuré par la société CLEARSY [Clearsy, 2006] [Clearsy, 2008]. SIEMENS (ex-MATRA) et ALSTOM ont participé à ce projet de la RATP [Behm, 1999] [Boulanger, 1999] [Boulanger, 2000]. Pour l'implémentation logicielle, le concept de "cycle en V" a été utilisé, qui inclut la conception et la validation du système. Les tests et la validation étaient dans la plupart des cas faits en double par le constructeur et par la RATP. La Méthode B était utilisée pour la modélisation et pour la validation de la logique, le langage d'implémentation était l'ADA [Lévi, 1988].

Il est à noter que :

- malgré les preuves formelles réalisées par le constructeur sur le logiciel du SAET[41] près de 400 situations « non sures » (non conformités pouvant conduire à une situation dangereuse – événement redouté) ;
- Ces situations « non sures » ont été révélées lors des essais réalisés par la maîtrise d'ouvrage (modélisations du fonctionnel sous RdP et scénarios automatisés)[42]

**Ces «oublis» de situations dangereuses découlaient naturellement de la difficulté d'identifier exhaustivement et d'écrire les propriétés de sécurité à vérifier (prédicats, obligations de preuve, postulats de fonctionnement).**

**La nature du langage n'a pas permis aux gens de métiers (experts signalisation) d'écrire voire d'approuver les propriétés prouvées par les constructeurs.**

### 5.2.5.2 B-„Werkstatt“

Die „B-Methode“ ist ein formales Softwareentwicklungsverfahren aus Frankreich für die abstrakte Modellierung des Programmverhaltens in der B-Sprache. Nach einer schrittweisen Verfeinerung erhält man ein konkretes Modell (das eine Untermenge der ADA oder C übersetzbaren Sprache darstellt). B [IMdR, 2009] wird verwendet für:

- die Realisierung korrekter Software durch Konstruktion.
- die Modellierung der Systeme in ihrem Umfeld.

Die Verfahren und Anwendungsprogramme werden von der Fa. Clearsy entwickelt [Clearsy, 2006] [Clearsy, 2008]. Siemens (ex-Matra) und Alstom haben sich an diesem RATP-Projekt [Behm, 1999] [Boulanger, 1999] [Boulanger, 2000] beteiligt. Für die Softwareimplementierung wurde das Konzept des „V-Zyklus“ angewandt; dieses umfasst den Entwurf und die Validierung des Systems. In den meisten Fällen wurden die Tests und die Validierung zweimal durchgeführt (vom Hersteller und von der RATP). Das B-Verfahren diente der Modellierung und der Validierung der Logik; ADA war die Implementierungssprache [Levi, 1988].

Ferner ist anzumerken, dass:

- es trotz der formalen Beweise des Herstellers bezüglich der SAET[43]-Software, ca. 400 „unsichere“ Situationen (Nichtkonformitäten, die zu einer gefährlichen Situation – einem befürchteten Ereignis führen können) gab.
- diese „unsicheren“ Situationen durch die Versuche des Auftraggebers entdeckt wurden (Modellierung der Funktionalitäten mit PN und automatische Szenarien)[44].

**Dieses „Vergessen“ von gefährlichen Situationen entstand natürlich aufgrund der Schwierigkeit, die zu prüfenden Sicherheitseigenschaften vollständig zu identifizieren und aufzuschreiben (Prädikate, Beweisauflagen, Anforderungen an die Funktionsweise).**

**Die Eigenschaften der Sprache haben es den Fachleuten (Signaltechnikexperten) nicht erlaubt, die von den Herstellern bewiesenen Eigenschaften aufzuschreiben bzw. zu genehmigen.**

[41] 20 dossiers de principe, 23 modèles, 3 versions des 3 applicatifs, 115000 lignes de code B => 90000 lignes de code ADA générées, 27800 obligations de preuves

[42] Validation sur 30 cahiers de tests, 5000 tests fonctionnels, 400 remarques critiques pour la sécurité, 110 anomalies sur les vérifications logicielles

[43] 20 Grundsatzdokumente, 23 Modelle, drei Versionen der drei Anwendungen, 115 000 Zeilen von B-Code ⇒ 90 000 Zeilen von ADA-Code, 27 800 Beweisauflagen

[44] Validierung mit 30 Testkatalogen, 5000 funktionalen Tests, 400 sicherheitskritische Anmerkungen und 110 Anomalien bei der Softwareüberprüfung

### 5.2.5.3 Constat actuel dans le ferroviaire

Indifféremment du langage de programmation utilisé, on observe une tendance marquée chez les « constructeurs » de matériels ferroviaires à l'utilisation de suites de modélisation telle que SCADE. La programmation "basée modèle" est plus aisée et intuitive. Ces suites permettent de vérifier la logique du logiciel au fur et à mesure de sa conception. La génération automatique du code permet de diminuer le "facteur humain" dans la phase de conception & codage des logiciels.

La «certificabilité affichée» du code généré est perçue comme un atout supplémentaire majeur. Ceci alors que les apports et les limites d'une telle certification de l'outil de génération de code ne sont pas clairs (sont pour le moins discutables : en quoi les contraintes temps réel de l'aéronautique correspondent à celles du ferroviaire et réciproquement).
Du côté des langages de programmation, on note l'utilisation du « vieux » ADA dans un grand nombre de projets. Ce n'est pas un langage mort. De nouveaux standards, de nouveaux compilateurs et de nouveaux environnements de travail sont toujours en cours d'élaboration. Néanmoins, même pour ce langage, il y a une tendance à utiliser des suites de modélisation avec la génération du code automatique.
De gros efforts sont entrepris pour créer des sous-ensembles de C, C++ et Java utilisables dans des systèmes critiques. Parmi eux, certains semblent aboutis, prenant en compte les contraintes du ferroviaire et de l'aéronautique. Ces ateliers reposent sur des modélisations et permettent effectivement de réduire les risques inhérents à la production des logiciels, notamment les cohérences entre entités.

Ces ateliers et suites ne répondent pas aux deux aspects que nous avons abordés :

- les spécifications reprennent-elles effectivement les expressions de besoins en matière de programme de fonctionnement et de propriétés de sécurité du système global ? Malgré ces méthodes des incidents graves sont survenus, peu ont été publiés ;
- le code exécuté est-il l'image rigoureuse des modèles qui ont été prouvés, notamment sur les aspects temps réels et concomitance des traitements ?

**Ainsi, si un pas est en train de se faire, il ne nous semble aller suffisamment loin : les aspects mathématiques coupent les ponts avec les aspects métiers.**

### 5.2.5.3 Heutiger Stand im Bahnbereich

Unabhängig von der verwendeten Programmiersprache, wird bei den „Herstellern" im Bahnbereich eine starke Tendenz zur Verwendung von Modellierungssprachen (beispielsweise SCADE) festgestellt. Die „modellorientierte Programmierung" ist leichter und intuitiver. Diese Programmpakete dienen der Prüfung der Softwarelogik während der Konzeption. Die automatische Codeerzeugung dient der Reduzierung des „Faktors Mensch" bei Softwareentwurf und Codierung.

Die „angegebene Zertifizierbarkeit" des erzeugten Code gilt als wesentlicher zusätzlicher Trumpf, obwohl die Vorteile und Grenzen einer solchen Zertifizierung des Codeerzeugungstools nicht klar sind (es ist zumindest strittig inwiefern Echtzeitrandbedingungen der Luftfahrt denen der Eisenbahn entsprechen und umgekehrt).

Seitens der Programmiersprachen stellt man bei zahlreichen Projekten die Verwendung der „alten" ADA-Sprache fest: sie ist keine tote Sprache. Neue Standards, neue Compiler und neue Arbeitsumgebungen entstehen, aber selbst bei dieser Sprache neigt man dazu, Modellierungsprogrammpakete mit automatischer Codeerzeugung zu verwenden. Es gibt viele Bemühungen, in kritischen Systemen verwendbare Teilmengen von C, C++ und Java zu entwickeln: einige davon scheinen erfolgreich zu sein und die Randbedingungen der Luftfahrt und der Eisenbahn zu berücksichtigen. Diese Entwicklungsumgebungen beruhen auf Modellierung und reduzieren tatsächlich die Risiken der Softwareherstellung, insbesondere bezüglich der Kohärenz zwischen den Einheiten.

Diese Entwicklungsumgebungen und Programmpakete geben aber keine Antwort auf die bereits erörterten Aspekte:

- Berücksichtigen die Spezifikationen tatsächlich den Bedarf bezüglich des Funktionsprogramms und der Sicherheitseigenschaften des Gesamtsystems? Trotz dieser Verfahren gab es schwere Störfälle und nur wenige davon wurden publik gemacht.
- Spiegelt der ausgeführte Code strikt die bewiesenen Modelle wieder, insbesondere im Hinblick auf Echtzeit und simultane Verarbeitung?

**Wenn auch ein Schritt in die richtige Richtung gemacht wird, so scheint dieser nicht weit genug zu sein: die mathematischen Aspekte trennen die Verbindung zu den Fachleuten.**

**C'est pourquoi notre travail propose une approche nouvelle qui pourrait, dans certains cas comme pour les postes d'aiguillage, répondre aux points précédents.**

## 5.2.6 Normes et méthodes formelles

Aussi la place des méthodes formelles dans les normes actuelles est très limitée. L'usage des méthodes formelles est néanmoins fortement recommandé pour les systèmes de plus haut niveau de sécurité (SIL4 de l'EN50129 pour le ferroviaire, DAL A de la DO178B pour l'aérien…). L'usage de méthodes formelles au moyen d'outils automatisant leur application permettrait de :

- simplifier le plan de sûreté de fonctionnement (notamment en traitant tout le logiciel avec la même rigueur quel que soit son niveau de criticité) ;
- permettre au dossier de sûreté de fonctionnement d'apporter la preuve que les fonctionnalités réalisées par les logiciels ne permettent jamais, dans le domaine des postulats pris en considération, d'atteindre un état dangereux. Le logiciel est alors dit correct ;
- rejouer l'ensemble des preuves lors de n'importe quelle modification du logiciel de base ou des paramètres ;
- ne faire aucune hypothèse de niveau de sécurité logicielle « acceptable », hypothèse en soit inacceptable pour des systèmes en service sans interruption sur de longues périodes de temps (cas des postes d'aiguillages par exemple).

L'application des méthodes formelles conduit à une réflexion amont sensiblement différente de celle évoquée par les normes traitant de la sûreté de fonctionnement du logiciel. Ainsi dans le premier cas une attention toute particulière doit être portée à l'écriture des propriétés de sécurité et des postulats définissant leur domaine de validité. Dans le second cas, il est demandé d'identifier (aussi exhaustivement que possible) les situations dangereuses qu'il conviendra de traiter [Monin, 1996].

En tout état de cause, les normes n'abordent que les aspects de développement du logiciel de base et de celui traitant de la non correction et la complétude du fonctionnel applicatif.

A cette fin il est uniquement fait mention de recommandations d'organisation et de procédures d'assurance qualité du logiciel chargés de couvrir les éventuels écarts de formalisation, de spécification et de traduction informatique de haut niveau.

**Aus diesem Grund wird in dieser Arbeit ein neuer Ansatz vorgeschlagen: in einigen Fällen (wie bei Stellwerken) kann dieser Ansatz die vorangegangenen Punkte berücksichtigen.**

## 5.2.6 Normen und formale Methoden

Der Stellenwert formaler Verfahren ist in den heutigen Normen sehr niedrig. Die Verwendung formaler Verfahren wird jedoch stark für Systeme mit höchster Sicherheitsstufe empfohlen (SIL4 aus der EN50129 für die Eisenbahn, DAL A der DO178B für die Luftfahrt…). Die Verwendung formaler Verfahren mit Hilfe von Werkzeugen, die ihre Anwendung automatisieren würde folgendes erlauben:

- Der Funktionssicherheitsplan wird vereinfacht (insbesondere durch die strikte Behandlung der gesamten Software, auf jeder Kritizitätsebene).
- Das Funktionssicherheitsdokument könnte auf diese Weise beweisen, dass die von der Software erfüllten Funktionen im Bereich der berücksichtigten Anforderungen niemals zu einer gefährlichen Situation führen. In diesem Fall gilt die Software als korrekt.
- Bei jeder Änderung der Grundsoftware oder der Parameter können alle Beweisführungen wiederholt werden.
- Es wird im Hinblick auf die „annehmbare" Softwaresicherheit keine Annahme gemacht; eine solche Annahme ist als solche nicht bei Systemen akzeptierbar, die länger ununterbrochen betrieben werden (Stellwerke zum Beispiel).

Die Anwendung formaler Verfahren führt zu Vorüberlegungen die sehr von den Vorüberlegungen abweichen, die in den Normen erwähnt werden, die die Funktionssicherheit von Software behandeln. Im ersten Fall muss dem Aufschreiben der Sicherheitseigenschaften und der Anforderungen zur Definition des Gültigkeitsbereiches eine besondere Aufmerksamkeit geschenkt werden. Im zweiten Fall müssen die zu behandelnden gefährlichen Situationen (so komplett wie möglich) identifiziert werden [Monin, 1996].
Auf jeden Fall behandeln die Normen lediglich die Entwicklungsaspekte der Grundsoftware und der Software zur Behandlung der Nichtkorrektur und der Vollständigkeit der Anwendungsfunktionen.

Deswegen werden nur Empfehlungen zur Organisation und Verfahren zur Qualitätssicherung der Software erwähnt, zur Abdeckung der etwaigen Unterschiede in der Formalgestaltung, in der Spezifizierung und in der IT-Umsetzung auf hoher Ebene.

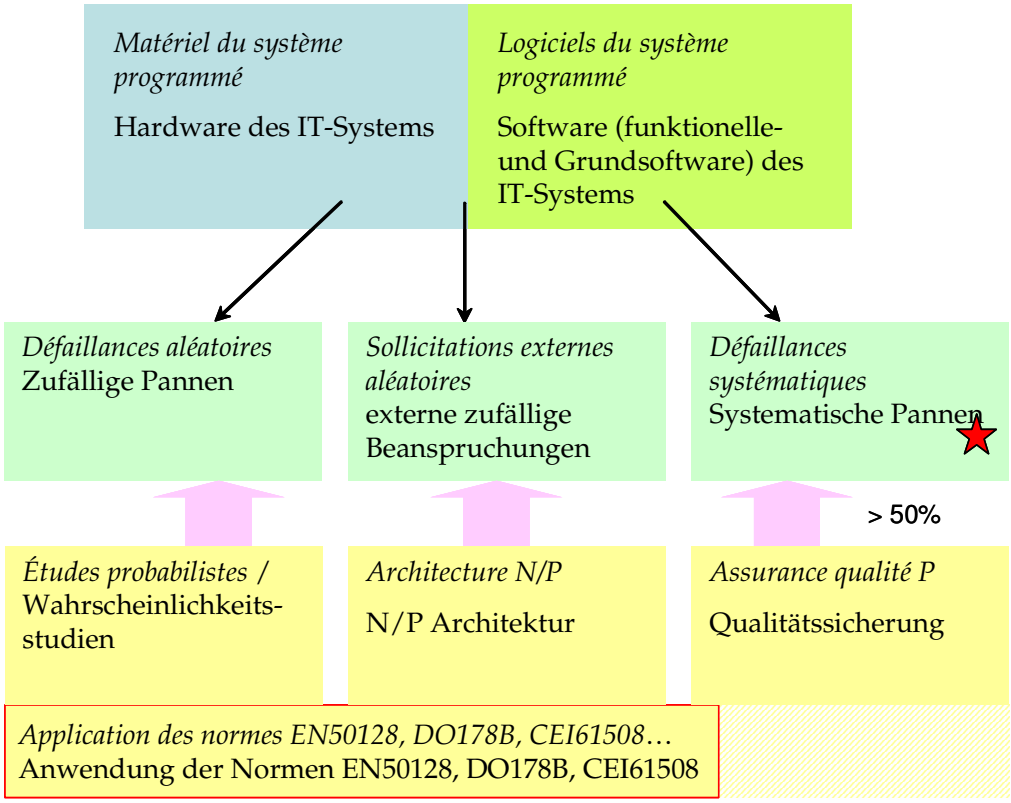



**Figure 5.2** : Correspondances avec les normes

**Abbildung 5.2**: Entsprechung mit den Normen

**Ce travail va proposer une approche permettant de couvrir l'ensemble du cycle de développement, des spécifications système à la preuve du code exécuté sur les machines cibles. Cette approche se veut applicable industriellement.**

**Il y a nécessité de bâtir un pont entre le monde universitaire et celui de l'industrie afin que l'application des méthodes formelles puisse être confiée à des gens de métier à même d'identifier les propriétés et les postulats requis.**

**Es wird in dieser Arbeit ein Ansatz vorgestellt, der den gesamten Entwicklungszyklus berücksichtigt, von der Systemspezifizierung bis hin zur Beweisführung des auf den Zielmaschinen ausgeführten Codes. Dieser Ansatz erhebt den Anspruch, auch industriell anwendbar zu sein.**

**Es muss eine Brücke geschlagen werden zwischen der akademischen und der industriellen Welt, damit die Fachleute selbst mit der Anwendung der formalen Verfahren beauftragt werden können: sie sollten in der Lage sein, die notwendigen Eigenschaften und Anforderungen zu identifizieren.**

## 5.3 Étude de sûreté système & Sécurité des systèmes programmés

Les systèmes industriels ne peuvent se limiter aux seuls systèmes informatisés. Se pose alors la difficulté d'assurer l'assemblage au sein d'une même étude des aspects [Bied, 1998] :

- probabilistes des études de sécurité systèmes ;
- déterministes des défauts de fonctionnement induits par des erreurs/incomplétudes des logiciels.

Ces deux approches sont complémentaires et peuvent s'alimenter l'une l'autre dans le cadre d'une démarche qui pourrait être la suivante :

1. Étude de sûreté système générale (disponibilité et sécurité) qui, outre ses résultats habituels, se doit :
   - d'identifier les fonctions critiques réalisées, totalement ou partiellement, par des systèmes informatisés ;
   - de définir les postulats de fonctionnement relatifs à ces fonctions, dans les différents profils de missions ;
   - de définir les propriétés de sécurité et fonctionnelles que doivent remplir ces fonctions (prédicats, obligations de preuve…)
2. Étude de la sécurité déterministe du système informatisé qui se doit :
   - de montrer que la fonction est réalisée de manière déterministe en toute circonstance ;
   - de chiffrer selon les méthodes classiquement retenues les taux de pannes sûre et non sûre imputables aux matériels supportant la fonction (en fonction de l'architecture notamment) ;
   - d'identifier la liste exhaustive des séquences ordonnées des entrées qui conduisent à un des états considérés soit comme non sûrs, soit comme sûrs. S'il n'en existe aucun, le taux de défaillance non sûre du fait du logiciel peut être considéré comme nul ;
   - valoriser les probabilités conditionnelles d'occurrence des différentes séquences afin d'estimer les taux de pannes sûres et non sûres liées aux sollicitations possibles des fonctions critiques.

## 5.3 Betriebssicherheitsanalyse und Sicherheit programmierter Systeme

Industrielle Systeme beschränken sich nicht auf IT-Systeme. Die Schwierigkeit besteht also darin, folgende Aspekte im Rahmen einer einzigen Analyse zu untersuchen [Bied, 1998]:

- Stochastische Aspekte der Systemsicherheits-analyse
- Deterministische Aspekte der Funktionsfehler aufgrund von Softwarefehlern und Unvoll-ständigkeit der Software.

Beide Ansätze ergänzen einander im Rahmen folgender Vorgehensweise.

1. Allgemeine Systembetriebssicherheitsanalyse (Verfügbarkeit und Sicherheit); außer den üblichen Ergebnissen muss diese Analyse folgendes erbringen:
   - Identifizierung der kritischen Funktionen, die vollständig oder teilweise von IT-Systemen realisiert werden
   - Definition der Anforderungen an die Funktionsweise dieser Funktionen je nach Aufgabenprofil
   - Definition der Sicherheitseigenschaften bzw. der Funktionen, die zu erfüllen sind (Prädikate, Beweispflicht…)
2. Untersuchung der deterministischen Sicher-heit des IT-Systems, mit folgenden Zielsetzungen:
   - Beweis der jederzeit deterministischen Durchführung der Funktion
   - Quantifizierung, gemäß der üblicherweise gewählten Verfahren, der Ausfallraten (sichere und unsichere Ausfälle) der die Funktion tragenden Hardware (und dies insbesondere in Bezug auf die gewählte Architektur)
   - Identifizierung der kompletten Liste der ordnungsgemäßen Eingabesequenzen, die zu einem als „unsicher" oder „sicher" geltenden Zustand führen. Falls es keine solchen Zustände gibt, gilt die Ausfallrate der unsicheren Ausfälle aufgrund der Software als null.
   - Bewertung der bedingten Wahrscheinlich-keiten der verschiedenen Sequenzen zur Abschätzung der Ausfallraten (sichere und unsichere Ausfälle), aufgrund möglicher Beanspruchungen der kritischen Funktionen.

3. Compléter l'étude de sûreté système générale (disponibilité et sécurité) en intégrant les taux d'occurrences des comportements sûrs et non sûrs des fonctions critiques. En fin d'étude il est nécessaire de vérifier les conditions de validité des postulats et propriétés utilisées pour le point 2. **Il est à noter que cette démarche n'est envisageable qu'avec une approche formelle du logiciel qui est en mesure de donner exhaustivement toutes les séquences qui mènent à des situations redoutées.**

Nous avons vu que ce n'est pas le cas général des méthodes formelles présentées. Il se trouve que ce sera le cas de la méthode explicité au chapitre suivant.

## 5.4 Ce qu'il faut retenir pour la suite du travail

A compléter par les éléments clés qu'il faudra considérer ultérieurement pour présenter notre méthode formelle et l'utiliser dans la suite du travail :

- Les méthodes formelles classiquement utilisées sont à la fois « universitaires » et non applicables sur des systèmes industriels existants (conception *ex nihilo*). Aucune ne permet un usage industriel par des gens du métier de l'application informatique.
- Les méthodes formelles classiquement avancée se limitent toujours à valider un « modèle » et non le code exécuté (il y a toujours une phase de transcodage en un langage informatique et de compilation en un langage exécutable) ;
- Il faut distinguer deux familles de méthodes formelles, celles de conception formelle (vision maîtrise d'œuvre) et celles de validation formelle (vision maîtrise d'ouvrage). Industriellement c'est la seconde famille qui est la plus attendue et la plus efficace. Le plus économique et efficace serait une interprétation du langage formel lui-même ;
- Quelle que soit la méthode formelle à utiliser, la problématique principale réside dans l'explicitation des obligations de preuve (propriétés de sécurité et postulats). Celle-ci n'est envisageable que par des experts du métier.

3. Vervollständigung der allgemeinen Systembetriebssicherheituntersuchung (Verfügbarkeit und Sicherheit) durch Integration der Häufigkeit des sicheren bzw. unsicheren Verhaltens der kritischen Funktionen. Am Ende der Untersuchung müssen die Gültigkeitsbedingungen der unter Punkt 2 aufgeführten Anforderungen und Eigenschaften überprüft werden. **Es ist anzumerken, dass diese Vorgehensweise nur mit einem formalen Softwareansatz und mit kompletter Angabe aller zu einem befürchteten Ereignis führenden Sequenzen in Erwägung gezogen werden kann.**

Es wurde gezeigt, dass dies bei den vorgestellten formalen Verfahren nicht allgemein gilt; bei dem im nächsten Kapitel vorgestellten Verfahren gilt dies jedoch.

## 5.4 Weiteres Vorgehen

Zur Vervollständigung der später zu betrachtenden Schlüsselelemente (zur Vorstellung des neuen formalen Verfahrens) und zur Verwendung im weiteren Vorgehen sollte folgendes im Gedächtnis behalten werden:

- Die üblicherweise verwendeten formalen Verfahren sind „akademisch". Sie lassen sich nicht auf bestehende industrielle Systeme anwenden (*ex nihilo* Konzeption). Keines der Verfahren ist industriell von Informatikfachleuten anwendbar.
- Die üblicherweise verwendeten formalen Verfahren beschränken sich immer auf die Validierung eines „Modells" und validieren nicht die Ausführungsaspekte (es gibt immer eine Phase der Codierung in eine IT-Sprache und der Kompilierung in eine ausführbare Sprache).
- Man muss unterscheiden zwischen zwei Familien formaler Verfahren: diejenige des formalen Konzepts (Vorstellung des Auftragnehmers) und diejenige der formalen Validierung (Vorstellung des Auftragnehmers). Für die industrielle Anwendung ist die zweite Familie die am stärksten benötigte und effiziente. Das Wirtschaftlichste und das Effizienteste wäre die Interpretation der formalen Sprache selbst.
- Welches formale Verfahren auch verwendet wird, das Hauptproblem ist die Erläuterung der Beweisauflagen (Sicherheitseigenschaften und Anforderungen): dies kann nur von einem Fachmann gemacht werden.

# CHAPITRE 6

# Une nouvelle méthode pour une validation formelle des systèmes informatiques critiques

## 6.1 Les besoins des postes modernes - Démarche de conception du Module d'enclenchement (MEI)

### 6.1.1 Traitement des défauts systématiques

Les chapitres précédents (3 et 4) ont permis de conceptualiser les principales attentes que tout gestionnaire d'infrastructure peut exprimer. Les caractéristiques du poste d'aiguillage idéal peuvent être résumées comme suit :

- Les aspects « fonctionnel applicatif » et « système informatique temps réel » avec leurs contraintes propres doivent pouvoir être traités distinctement, l'un indépendamment de l'autre. Ainsi, une évolution fonctionnelle du poste ne doit pas requérir une évolution et/ou une revalidation sécurité du système temps réel (matériel et logiciel). Une évolution système temps réel ne doit pas requérir de modification du fonctionnel applicatif (fonctions de signalisation) ;
- L'expression du « fonctionnel applicatif » associé à l'environnement du poste (réglementaire, humain et topologique) doit pouvoir être validé en totalité, formellement, dans les meilleures conditions possibles et par des agents du métier de la signalisation ferroviaire ;
- Le langage de description du «fonctionnel applicatif» (fonctionnalités) doit permettre une compréhension par les agents du métier de la signalisation (études, maintenance, essais, exploitation), une interprétation totalement déterministe par le système temps réel. Un langage spécifique (dit AEFD) a été défini à ces effets ;

# KAPITEL 6

# Neue Methode für eine formale Validierung kritischer IT-Systeme

## 6.1 Bedarf bei modernen Stellwerken – Vorgehensweise bei der Entwicklung des Sicherungsmoduls (MEI)

### 6.1.1 Bearbeitung systematischer Fehler

In den vorangehenden Kapiteln (3 und 4) wurden die Haupterwartungen eines jeden Infrastrukturbetreibers erörtert. Die Eigenschaften eines idealen Stellwerkes können wie folgt zusammengefasst werden:

- Die funktionellen Aspekte und die Aspekte des Echtzeit-IT-Systems, mit den jeweils spezifischen Randbedingungen, müssen einzeln und unabhängig behandelt werden können. Eine funktionelle Weiterentwicklung des Stellwerks darf keine Weiterentwicklung oder erneute Sicherheitsüberprüfung des Echtzeitsystems erfordern. Die Weiterentwicklung des Echtzeitsystems darf keine Änderung der funktionellen Software erfordern.
- Die Beschreibung der funktionellen Software in Verbindung mit dem Stellwerksumfeld (Vorschriften, Betriebspersonal, Topologie) muss, unter den bestmöglichen Bedingungen und von Signaltechnikexperten, komplett formal überprüft werden können.
- Die Beschreibungssprache der „funktionellen Software“ (Funktionen) muss von den Signaltechnikexperten (Entwicklungen, Wartung, Tests, Betrieb) verstanden werden. Sie muss auch komplett deterministisch durch ein Echtzeitsystem interpretierbar sein. Zu diesem Zweck wurde eine spezifische Sprache („AEFD“ genannt) definiert.

- L'architecture système temps réel doit être telle que tout aléa dans l'interprétation ou l'exécution d'une fonctionnalité se traduise par la mise en position de repli sûre de tout le poste d'aiguillage ;
- Afin de conserver le lien avec ce passé (retour d'expérience de 150 ans d'exploitation des chemins de fer…), le « fonctionnel applicatif » décrit sous forme d'automates écrits en langage AEFD, doit conserver les fonctionnalités et leur organisation en couches fonctionnelles unifiées des postes d'aiguillage à itinéraires : le changement de technologie (le comment) n'a pas à remettre en cause les fonctions (le pourquoi). Le poste reprend les fonctionnalités du PRS pour la partie enclenchement et celles du PRCI pour la partie supérieure. Les différents types de poste sont expliqués dans l'annexe B.5.

Les systèmes informatiques critiques présentent des difficultés de conception et de validation. Les modes de défaillance ne sont pas connus. Les paramètres ayant un réel impact sur l'événement redouté majeur sont difficiles à identifier. L'évaluation de ces paramètres repose sur une analyse déterministe distinguant les défauts systématiques et les conditions de déclenchement correspondantes (cf. Figure 6.1).

- Die Architektur des Echtzeitsystems muss derart sein, dass jedes zufallsbedingte Ereignis bei der Interpretation oder Ausführung einer Funktion zur Sicherheitsstellung des gesamten Stellwerks führt.
- Um die Verbindung mit der Vergangenheit zu bewahren (150 Jahre Eisenbahnbetriebserfahrung), muss die funktionelle Software, die in Form von Automaten in der AEFD-Sprache beschrieben ist, alle Funktionen und deren Organisation in einheitlichen Funktionsebenen des Fahrstraßenstellwerks beibehalten: eine Änderung der Technologie darf die Funktionen nicht in Frage stellen. Das Stellwerk beinhaltet bezüglich des Verschlussteils die PRS-Funktionen, und bezüglich der höheren Ebene die PRCI-Funktionen. Die verschiedenen Stellwerkstypen werden im Anhang B.6 beschrieben.

Bei kritischen IT-Systemen gibt es Entwurfs- und Validierungsschwierigkeiten. Die Ausfallmodi sind nicht bekannt. Die Parameter, die einen tatsächlichen Einfluss auf befürchtete Ereignisse haben, sind schwer zu identifizieren. Die Bewertung dieser Parameter beruht auf einer deterministischen Analyse, die systematische Fehler und die entsprechenden Auslösebedingungen unterscheidet (Abb. 6.1).

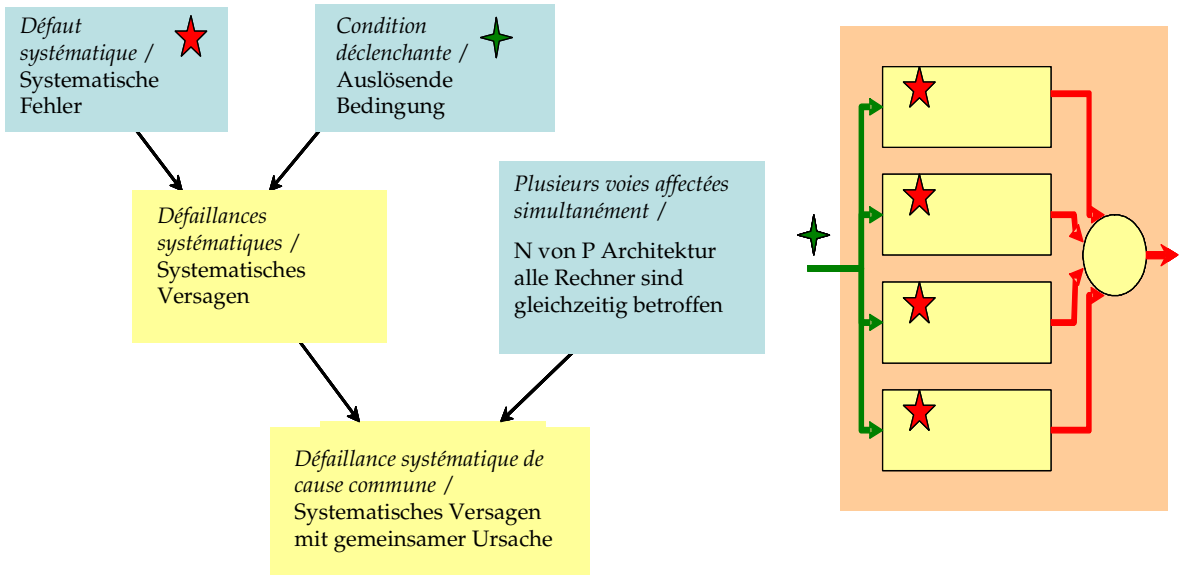



**Figure 6.1** : Principes de l'Analyse Déterministe / **Abbildung 6.1**: Prinzipien der deterministischen Analyse

Il est nécessaire d'opérer par conception à la réduction des défauts systématiques et à la réduction des conditions d'activation. Par exemple, il s'agit d'identifier les facteurs qui peuvent amener le système à s'écarter des chemins suivis en situations normales ou suivis occasionnellement mais bien testés.

Es ist bei der Konzeption wichtig, die systematischen Fehler und die Auslösebedingungen zu reduzieren. Man identifiziert zum Beispiel die Faktoren, in denen das System von der Normalsituation oder einer selten vorkommenden, aber sorgfältig getesteten Situation abweichet.

***Domaine de fonctionnement de l'unité programmée* - Funktionsbereich des IT-Systems**

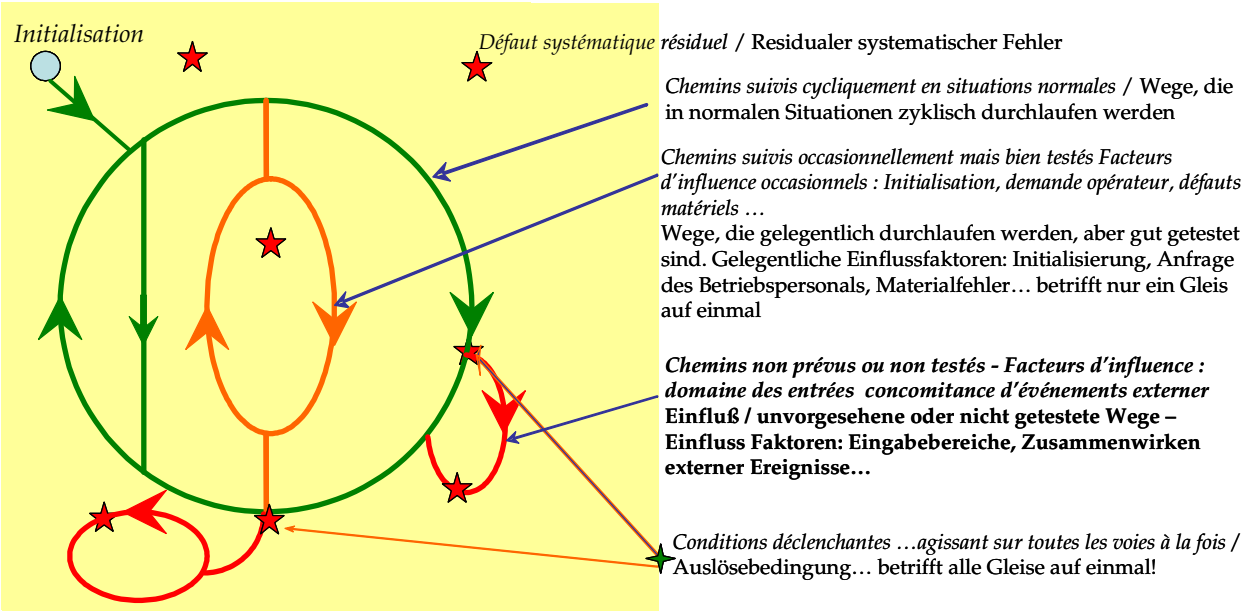



**Figure 6.2** : Principes de l'Analyse Déterministe - **Abbildung 6.2**: Prinzipien der deterministischen Analyse

Comme vu au Chapitre 3, la plus grande partie du risque des systèmes sûrs provient clairement non des «défaillances aléatoires»[45] mais de «défaillances systématiques».[46]

Il s'agit essentiellement, de défaillances communes impactant simultanément plusieurs équipements d'un système redondé, de défaillances systématiques des logiciels dues à des erreurs de conception (mettant en échecs les stratégies de diversification matérielle) et les défaillances dues à des séquences d'entrées mettant en défaut les barrières de sécurité.

L'état de l'art montre qu'il est très difficile, voire impossible, d'intégrer des modèles détaillés des systèmes informatiques dans un modèle probabiliste plus général.

Wie bereits im Kapitel 3 gezeigt, rührt bei sicheren Systemen der größte Teil der Risiken ganz klar vom „systematischen Versagen"[47] her und nicht vom „zufallsbedingten Versagen"[48].

Es handelt sich um gemeinsames Versagen, das gleichzeitig verschiedene Rechner eines redundanten Systems beeinflusst, um systematische Softwarefehler, die auf Konzeptionsfehlern beruhen (die die Strategie der Hardwarediversifikation zum Versagen bringen) und um Versagen aufgrund von Eingangssequenzen, die die Sicherheitsbarrieren außer Kraft setzen.

Der heutige Stand der Technik macht es sehr schwierig, wenn nicht unmöglich, detaillierte Modelle eines IT-Systems in ein allgemeines stochastisches Modell zu integrieren.

[45] L'informatique temps réel dans le domaine des systèmes critiques bénéficie d'un retour d'expérience de près de 30 ans.

[46] Ce même constat a déjà été fait du temps des postes mécaniques [Descubes, 1898]. La méthode Descubes appliquée systématiquement en France depuis 1900 a permis d'éradiquer ces erreurs systématiques dans les tables d'enclenchement

[47] Im Bereich der kritischen Systeme verfügt die Echtzeit-IT über eine ca. 30-jährige Erfahrung.

[48] Das war auch der Fall mit Mechanischen Stellwerke [Descubes, 1898]. Die Descubes Methode ermöglich diese systematische Fehlern zu absoluter Weise zu vermeiden

Pour répondre aux difficultés propres aux systèmes critiques à haut niveau de sécurité, les étapes suivantes ont été pensées et réalisées :

1. Comment réaliser un poste d'aiguillage où :
   - le fonctionnel applicatif peut être assimilé rigoureusement à un «automate à nombre d'état fini», forme d'automate que l'on sait prouvable *a priori* (processus combinatoire et séquentiel) ;
   - le fonctionnel applicatif est interprété de manière déterministe dans les conditions réelles de mise en œuvre ferroviaire ;
2. Quelle preuve réaliser et comment la justifier mathématiquement dans les conditions précédemment définies ;
3. Comment formaliser et exprimer les propriétés de sécurité et les postulats de fonctionnement ;
4. Comment automatiser la réalisation de cette preuve de validation sur les fonctionnels applicatifs réellement installés par les postes d'aiguillages de type PIPC (modules d'enclenchement (MEI) plus précisément).

## 6.1.2 Principes retenus pour la conception du poste PIPC

Les réseaux de PETRI ont été retenus comme support de spécification pour leurs propriétés mathématiques et leur possibilité d'expression (facilité d'appropriation par les experts métier). Pour une minimisation du coût des études de sécurité la distinction a été faite, par la mise en œuvre de deux types d'études de sécurité, entre les risques probabilistes et les risques déterministes :

- preuve formelle de spécification: Démonstration formelle de la complétude et de la suffisance des spécifications ;
- preuve de fonctionnement: Démonstration quantifiée de l'atteinte des objectifs de sécurité et FMD (graphes de Markov - indépendance de la sécurité vis à vis du matériel).

### 6.1.2.1 Architecture du PIPC

Parmi, les difficultés essentielles de conception des postes informatiques notons l'identification de toutes les solutions dégradées possibles (elles sont très nombreuses et peuvent concerner à la fois le poste lui-même et ses équipements externes), le besoin de sauvegarder en permanence la mémoire des états des installations pour permettre la réinitialisation du système sans créer de situation dangereuse et la «preuve» d'un fonctionnement conforme aux objectifs de sécurité.

Um Lösungen für die Schwierigkeiten kritischer IT-Systeme mit hohem Sicherheitsniveau zu finden, sind folgende Etappen durchdacht und verwirklicht worden:

1. Wie soll man ein Stellwerk aufbauen, bei dem:
   - die Funktionen strikt mit einem „endlichen Automaten" vergleichbar sind? Von diesen Automaten weiß man, dass sie beweisbar sind (kombinatorisches und sequenzielles Verfahren).
   - die Funktionen unter den tatsächlichen Bahnbetriebsbedingungen deterministisch interpretiert werden?
2. Welcher Beweis soll durchgeführt werden und wie kann man die zuvor erwähnten Bedingungen mathematisch rechtfertigen?
3. Wie kann man die Sicherheitseigenschaften und die Anforderungen formal gestalten und ausdrücken?
4. Wie kann der Validierungsbeweis der auf einem PIPC-Stellwerk tatsächlich installierten Funktionen (genauer gesagt auf dem Sicherungsmodul MEI) automatisiert werden?

## 6.1.2 Prinzipien des Entwurfs eines PIPC-Stellwerks

Interpretierbare Petrinetze als Spezifizierungsträger wurden aufgrund ihrer mathematischen Eigenschaften und ihrer Ausdrucksmöglichkeit gewählt (einfache Aneignung durch die Fachleute). Für eine Minimierung der Kosten der Sicherheitsstudien wurde eine Trennung zwischen stochastischen Risiken und deterministischen Risiken (durch die Umsetzung von zwei Arten von Sicherheitsnachweisen) durchgeführt:

- formale Spezifikationsüberprüfung: formeller Vollständigkeits- und „Hinlänglichkeits"-beweis der Spezifikationen
- Funktionsvalidierung: quantifizierter Beweis des Erreichens der Sicherheitszielsetzungen und FMD (Markovgraphen - Sicherheit unabhängig von den Maschinen).

### 6.1.2.1 Architektur des PIPC-Stellwerks

Die Identifizierung aller Rückfallebenen (diese Unterschied sind zahlreich und können sowohl das Stellwerk als auch dessen externe Anlagen betreffen) gehört zu den Hauptschwierigkeiten beim Entwurf eines IT-Stellwerkes; das gilt auch für die Notwendigkeit, ständig alle Anlagenzustände zu speichern, damit ein Systemneustart nicht zu einer gefährlichen Situation führen kann, sowie für den „Beweis", das der Betrieb den Sicherheitszielsetzungen entspricht.

La mise en œuvre pratique de cette méthode nécessite de disposer d'une machine cible déterministe et sûre (SIL4) qui interprète les spécifications prouvées. Lors de la conception du module d'enclenchement de la machine cible, il a été nécessaire d'intégrer les contraintes de validité de la preuve (automate à nombre fini d'états) et de la validité industrielle de la machine cible.

Le PIPC est un poste d'aiguillage informatique réalisant les fonctions d'enclenchements sous la conduite d'un poste de contrôle/commande et pilotant les objets terrain (aiguilles, signaux…). Ces fonctions d'enclenchement associées aux procédures d'exploitation contribuent à la couverture des risques ferroviaires.

L'ensemble du système s'intègre dans un environnement ferroviaire organisé selon les niveaux hiérarchique suivants :

Le niveau contrôle commande (N0) : ce niveau est pris en charge par le système de commande / contrôle MISTRAL[49] ou MADPI[50]. Ce système, est associé à un dispositif mural de visualisation dit Tableau de Contrôle Optique (TCO). Il constitue l'interface homme machine destiné à l'exploitant. Il assure les fonctions de commande et de contrôle des itinéraires, des protections, ainsi que des fonctions d'aide à l'exploitation de plus haut niveau (suivi des circulations, programmation automatique des itinéraires…).

Le niveau enclenchement (Niveau 1) : Il assure le passage en sécurité des trains, commande les appareils de voie dans la position requise par les itinéraires, en gère les enclenchements, l'espacement des circulations, les annonces aux passages à niveaux et en commande les signaux.

Le niveau (Niveau 2) d'interfaçage avec les équipements en campagne ou objets: il s'agit des produits NS1 (relais de sécurité) assurant les échanges d'informations entre les équipements en campagne et le niveau d'enclenchement.

Le niveau campagne (Niveau 3) : il s'agit des équipements physique comme par exemple les aiguilles, les signaux, les points de comptage d'essieux, les relais… soit l'ensemble des objets pilotés par le poste d'aiguillage.

Le PIPC assure les fonctions du niveau d'enclenchement (N1). Il met en œuvre des principes d'enclenchement nécessaires à l'exécution des commandes du niveau contrôle commande (N0) et renvoie les informations nécessaires à l'animation de l'interface homme machine destiné à l'exploitant.

Zur praktischen Implementierung dieses Verfahrens braucht man eine deterministische und sichere Zielmaschine (SIL4), welche die bewiesenen Spezifikationen interpretiert. Beim Entwurf des Sicherungsmoduls der Zielmaschine müssen die Gültigkeitsgrenzen des Beweises (endlicher Automat) und die Grenzen der industriellen Gültigkeit der Zielmaschine integriert werden.

Das PIPC-Stellwerk ist ein IT-Stellwerk, das mit Hilfe eines Steuer-/Kontrollstellwerks die Verschlussfunktionen ausführt und die Objekte vor Ort (Weichen, Signale…) steuert. In Verbindung mit den Betriebsvorschriften tragen diese Sicherungsfunktionen zur Vermeidung der Bahnrisiken bei.

Das ganze System ist in ein Bahnumfeld mit folgenden hierarchischen Ebenen integriert:

Steuerebene (Ebene 0): diese Ebene untersteht dem MISTRAL[51] oder dem MADPI[52] Steuersystem. Diese Systeme sind mit einer an der Wand befestigten, optischen Kontrollanzeige (TCO) verbunden. Diese Systeme bilden die Mensch-Maschine Schnittstelle mit dem Betriebspersonal. Sie sichern die Steuerung und die Kontrolle der Fahrstraßen, die Fahrwegsicherung, sowie die Betriebsunterstützungsfunktion (Zuglaufverfolgung, automatische Fahrstraßenprogrammierung…).

Sicherungsebene (Ebene 1): Sie gewährleistet die Sicherheit der Züge. Sie bringt die Weichen in die erforderliche Stellung. Sie verwaltet die Sicherungen, die Abstände zwischen den Zügen, die Vormeldungen an den Bahnübergängen und die Signalsteuerung.

Schnittstellenebene (Ebene 2) mit den externen Anlagen und Objekten: es handelt sich um NS1-Geräte (modulare Sicherheitsblockrelais) für den Informationsaustausch zwischen den Anlagen vor Ort und der Sicherungsebene.

Örtliche Ebene (Ebene 3): Es handelt sich um physische Elemente wie z. B. Gleisanlagen, Weichen, Signale, Achszähler, Relais, usw., kurzum, alle vom Stellwerk gesteuerten Objekte.

Das PIPC-Stellwerk sichert die Funktionen (Ebene 1). Es wendet die Sicherungsprinzipien an, die zur Ausführung der Steuerungen auf der Anforderungs- und Überwachungebene (Ebene 0) notwendig sind und gibt die für die Mensch-Maschineschnittstelle des Bedienpersonals notwendigen Informationen zurück.

[49] Ensemble de Modules Informatiques de Signalisation, de Transmission et d'Alarmes apte à piloter de nombreux types de postes d'aiguillage, dont le PIPC

[50] Module d'exploitation local spécifique du PIPC

[51] IT-Signalisierungmodule, die geeignet sind, zahlreiche Typen von Stellwerken zu steuern, darunter das PIPC-Stellwerk

[52] örtliches Betriebsmodul des PIPC-Stellwerks

L'architecture du poste d'aiguillage PIPC a été volontairement conçue sur trois couches :

- du matériel non spécifique au monde ferroviaire : ce sont des PC dans un conditionnement industriel répondant aux normes CEI1004.4 (architecture 2/2 pour la sécurité, doublé pour la disponibilité) ;
- un logiciel de base indépendant du matériel et du fonctionnel. Il gère les ressources informatiques (entrées, sorties, messages, mémoires, communication, temporisations, archivage…) et interprète le fonctionnel applicatif en utilisant des règles imposées. Ce logiciel de base constitue une structure d'accueil (logiciel commun à tous les fonctionnels paramétrés des PIPC). Il comporte en particulier le moteur d'exécution pour la résolution des enclenchements modélisés en graphes (exécution des spécifications fonctionnelles modélisées en graphes écrits en langage AEFD) ;
- un logiciel applicatif pour la définition des enclenchements (graphes) et les interfaces avec le poste de contrôle/commande. Ce fonctionnel applicatif (principes de signalisation utilisables sur un ensemble de lignes d'un réseau ferroviaire) repose sur des «graphes d'état génériques asynchrones» paramétrés pour chaque site.

Dans la suite, le terme «d'interpréteur» sera utilisé pour désigner le logiciel du deuxième niveau qui interprète les «réseaux de Petri fonctionnels». (Figure 6.4)

### 6.1.2.2 Articulation Matériel & Logiciel

L'articulation Matériel et Logiciel retenue doit permettre de maîtriser les coûts de développement et d'améliorer la durée de vie par rapport aux générations informatiques précédentes.

Ces réseaux de Petri interprétables sont des graphes fonctionnels décrits dans un fichier ASCII[53] et interprétés en temps réel par l'interpréteur de la machine cible. (cf. paragraphe 6.2.4)

Les règles garantissent :
- un fonctionnement déterministe de l'interpréteur ;
- une compréhension non équivoque (l'écriture, la lecture, la modification possible par tous les agents du métier de la signalisation…) des automates ainsi définis.

Die Architektur des PIPC-Stellwerks beruht absichtlich auf drei Bestandteilen:

- nicht bahnspezifische Hardware: es handelt sich um industrielle PCs gemäß den Normen CEI1004.4 (Architektur 2/2 für die Sicherheit, verdoppelt für die Verfügbarkeit).
- von der Hardware und von den Funktionen unabhängige Grundsoftware. Sie verwaltet die IT-Ressourcen (Eingänge, Ausgänge, Speicher, Kommunikation, Verzögerungszeiten, Archivierung…) und interpretiert die Funktionen unter Verwendung der auferlegten Regeln. Diese Grundsoftware ist eine „Empfangsstruktur" (für alle PIPC-Stellwerke identische Software). Sie umfasst insbesondere Löser für die als Graphen modellierten Sicherungsfunktionen (Ausführung der funktionellen Spezifikationen, die durch in der AEFD-Sprache geschriebene Graphen modelliert werden).
- Anwendungssoftware für die Definition der Sicherungen (Graphen) und die Schnittstellen mit der Fernsteuerung. Diese Funktionen (Signalgebungsprinzipien, die auf einer Reihe von Bahnstrecken verwendet werden können) beruhen auf „asynchronen und generischen Graphen", die für jedes Stellwerk parametrisiert werden.

Im Folgenden wird der Begriff „Interpreter" benutzt, um die Software der zweiten Ebene zu bezeichnen, die die funktionellen Petrinetze interpretiert. (Abb. 6.4)

### 6.1.2.2 Hardware-Software Verbindung

Mit der gewählten Hardware-Software Verbindung sollen die Entwicklungskosten reduziert und die Lebensdauer im Vergleich zu den früheren IT-Generationen verlängert werden.

Die interpretierbaren Petrinetze sind funktionelle Graphen, die in einer ASCII[54]-Datei definiert und vom Interpreter der Zielmaschine in Echtzeit interpretiert werden. (siehe Abschnitt 6.2.4)

Die Regeln gewährleisten folgendes:
- eine deterministische Funktionsweise des Interpreters
- ein eindeutiges Verständnis (Schreiben, Lesen, mögliche Änderung durch Signaltechnikexperten…) der auf diese Weise definierten Automaten.

[53] Ce fichier ASCII est simplement transformé en fichier binaire pour faciliter cette interprétation temps réel

[54] Diese ASCII-Datei wird lediglich zur Erleichterung der Echtzeitinterpretation in eine binäre Datei umgewandelt.

Nous avons volontairement conçu le PIPC de manière à intégrer les contraintes nécessaires à la réalisation ultérieure d'une preuve, à savoir :

- les spécifications fonctionnelles (les réseaux de Petri asynchrones écrits en langage AEFD) sont directement interprétées, sans être réécrites ;
- le traitement d'un seul événement est traité à la fois (ceci ne pose pas de problème de garantie de traitement, même en cas d'avalanche d'événement), la chronologie des événements externes est ainsi conservée en toutes circonstances[55] ;
- les règles de gestion des temporisations sont définies rigoureusement (hors graphes).

Das PIPC-Stellwerk wurde absichtlich so entworfen, dass die für die spätere Beweisführung notwendigen Randbedingungen integriert sind:

- Die funktionellen Anforderungen (Petrinetze in AEFD-Sprache) werden direkt interpretiert und nicht umgeschrieben.
- Kein einziges Ereignis wird gleichzeitig mit anderen bearbeitet (darum gibt es kein Problem mit der Bearbeitungsgarantie, selbst im Falle einer Ereignislawine); auf diese Weise wird die Chronologie der externen Ereignisse immer beibehalten[56].
- Die Regeln zur Verwaltung der Verzögerungszeiten werden strikt definiert (außerhalb der Graphen).

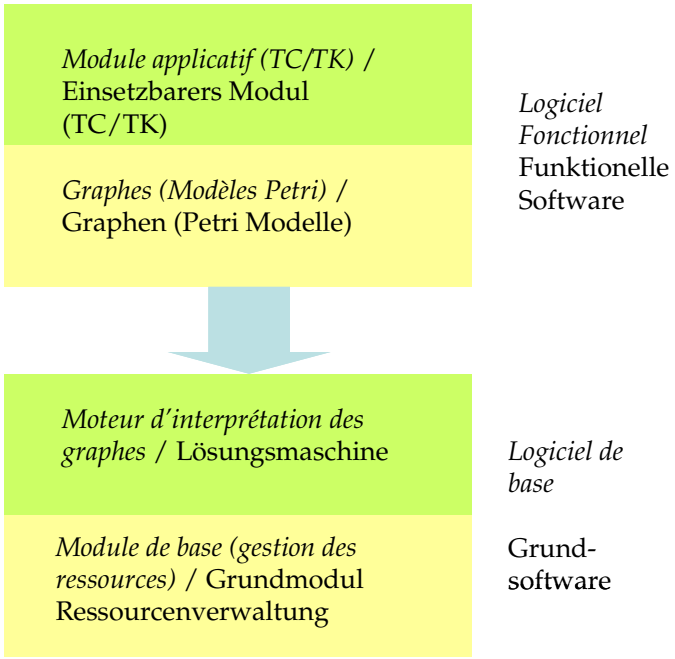



**Figure 6.3** : Principe de l'architecture Hard Soft du PIPC / **Abb. 6.3:** Hardware-Software Prinzip des PIPC

Les différentes fonctions du logiciel de base peuvent être illustrées par la figure 6.4.

Die verschiedenen Funktionen der Grundsoftware werden in der Abb.6.4 illustriert.

[55] En cas d'impossibilité, une fonction de sécurité provoque la mise en position de repli permanente et sûre toutes les sorties du poste et le poste lui-même.

[56] Im Falle eines Konflikts schaltet eine Sicherheitsfunktion alle Stellwerksausgänge, sowie das Stellwerk selbst auf die ständige und sichere Rückfallebene um.

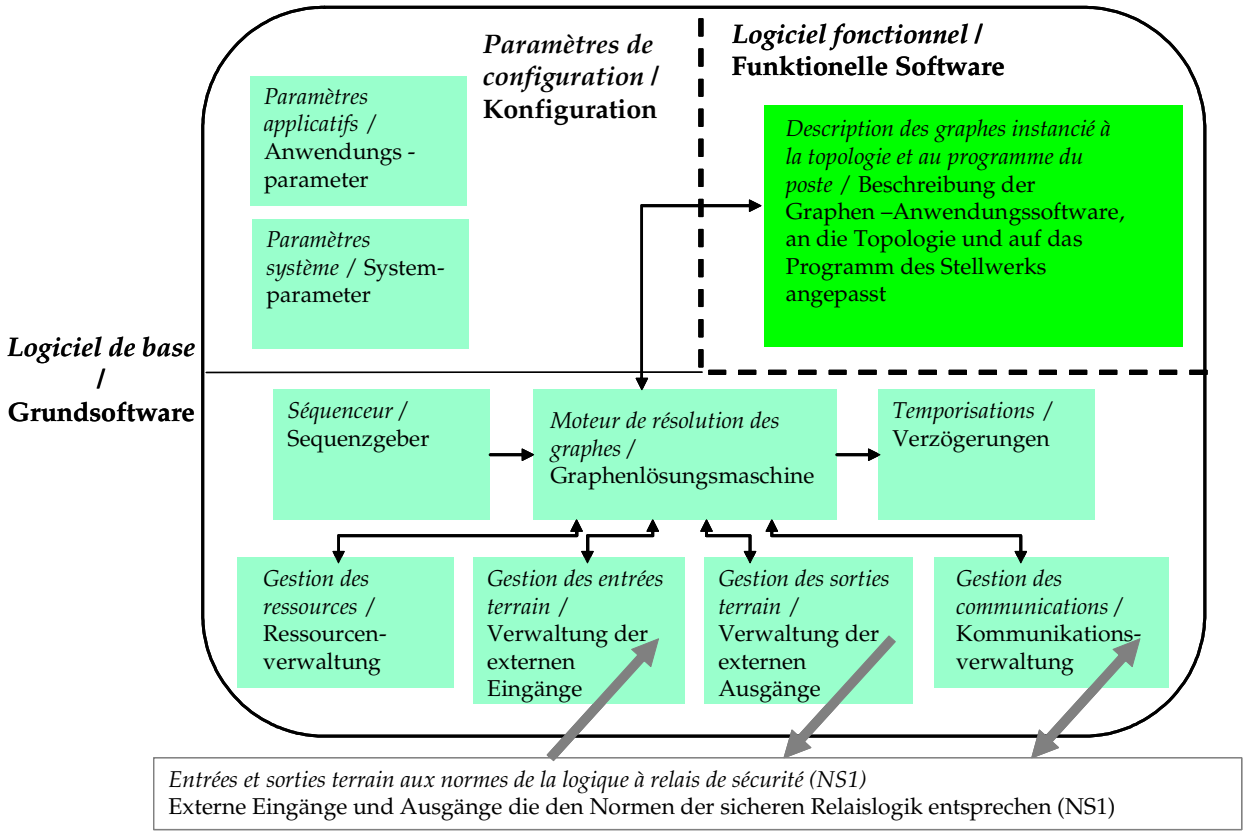



**Figure 6.4** : Architecture logicielle du MEI / **Abbildung 6.4**: Softwarearchitektur des MEI

D'une manière générale, dans un système temps réel équipé d'ordinateurs, il est nécessaire de différencier différents aspects:

- Les fonctionnalités que le système doit réaliser. Dans ce cas, le problème principal est le suivant: les spécifications et leurs transformations dans le code final acquis sont elles correctes à 100% ?
- L'ensemble matériel et logiciel de base réalise les fonctionnalités temps réel du système. Dans ce cas, le problème principal est le suivant: le taux horaire de défaillance non sûre résiduel ciblé peut-il être garanti par l'architecture dans les conditions d'usage réelles ?

Concernant le premier aspect, il est clair qu'une démarche de « développement en qualité » ne peut répondre efficacement à notre problématique : les fonctionnalités sont elles "correctes" ? *A priori* une approche formelle couvrant l'ensemble du cycle de développement peut y répondre favorablement (du programme fonctionnel en amont des spécifications au code interprété par la machine cible). C'est dans cet esprit qu'a été conçu le poste PIPC comme nous le verrons plus loin.

Im Allgemeinen muss man in einem IT-Echtzeitsystem verschiedene Aspekte trennen:

- die vom System zu erfüllenden Funktionen. Das Hauptproblem ist in diesem Fall folgendes: sind die Spezifikationen und deren Umwandlung in den Endcode zu 100 % korrekt?
- Die Hardware und die Grundsoftware führen zusammen die Echtzeitfunktionen des Systems aus. In diesem Fall ist das Hauptproblem folgendes: kann die Architektur unter den tatsächlichen Bedingungen die stündliche Zielrate der nicht sicheren Restfehler gewährleisten?

Beim ersten Aspekt ist es klar, dass die „Entwicklung mit Qualitätskontrolle" das Problem nicht effizient lösen kann: sind die Funktionen „korrekt"? *A priori* kann ein formaler Ansatz, der den gesamten Entwicklungszyklus abdeckt, darauf positiv antworten (vom funktionellen Programm vor den Spezifikationen bis hin zum von der Zielmaschine interpretierten Code). Das PIPC-Stellwerk wurde in diesem Sinne entworfen, wie später noch gezeigt wird.

#### 6.1.2.3 Temps de réponse du PIPC

Le temps de réponse du PIPC est inférieur à 1 seconde (délai maximal entre le changement d'une entrée ou la réception d'un message et la commande d'une sortie terrain conséquemment). Les changements d'état supérieurs à 50 ms sont détectés et prise en compte par le PIPC (génération d'événements). Le temps de cycle de traitement est d'un changement d'état est de 100 ms. Des études SNCF ont montré que le nombre moyen d'entrées d'un module d'enclenchement pouvant changer d'état est de 7 par seconde.
D'une manière générale, les conditions de gestions des entrées et des sorties terrain du poste répondent aux conditions techniques de la logique à relais de sécurité SNCF (postulat de fonctionnement).

**Dans ces conditions, le module d'enclenchement est une machine strictement déterministe. Il permet de :**

- **réaliser des fonctions de signalisation avec un très haut niveau de sécurité (SIL4) ;**
- **bien différencier les risques probabilistes et déterministes. Ceci permet d'apporter des solutions simples et adaptées à nos besoins de maintien en condition opérationnelle sur de grands temps.**

#### 6.1.2.3 Antwortzeit des PIPC-Stellwerks

Die Antwortzeit des PIPC liegt immer unter einer Sekunde (maximale Zeit zwischen der Änderung eines Eingangs oder dem Empfang einer Meldung und der entsprechenden Ansteuerung vor Ort). Zustandsänderungen, die länger als 50 ms brauchen, werden vom PIPC immer entdeckt und berücksichtigt (Erzeugung von Ereignissen). Die Dauer des Bearbeitungszyklus einer Zustandsänderung beträgt 100 ms. SNCF-Studien haben gezeigt, dass die durchschnittliche Anzahl von Eingängen eines Sicherungsmoduls, dessen Zustand sich ändern kann, 7 pro Sekunde beträgt.

Im Allgemeinen entsprechen die Bedingungen der Ein- und Ausgangsverwaltung des Stellwerkes den technischen Bedingungen der SNCF-Logik mit Sicherheitsrelais (funktionelle Anforderungen).

**Unter diesen Bedingungen ist das Sicherungsmodul eine strikt deterministische Maschine. Diese erlaubt:**

- **eine Ausführung der Signalisierungsfunktionen mit einer sehr hohen Sicherheit (SIL4).**
- **eine gute Trennung der stochastischen und der deterministischen Risiken. Dadurch erhält man einfache und bedarfsentsprechende Lösungen mit einer hohen Betriebsbereitschaft.**

## 6.2 Architecture générique du PIPC

### 6.2.1 Architecture matérielle du MEI - Construction de la sécurité

D'autres propriétés ont été introduites lors de la conception du PIPC afin de :

- réduire les risques probabilistes de modes communs par une triple hétérogénéité :
  - matérielle (PC différents, BIOS différents, compatibilité CEM selon la norme CEI1004 Niveau 4) ;
  - logicielle (deux unités de traitement utilisant un compilateur différent C) ;
  - temporelle (synchronisme maître esclave, entrelacement des traitements applicatifs et autotests…) ;

## 6.2 Allgemeine Architektur des PIPC Stellwerks

### 6.2.1 Hardwarearchitektur des MEI – Sicherheitskonstruktion

Weitere Eigenschaften wurden bei der Entwicklung des PIPC berücksichtigt:

- Reduzierung der gemeinsamen Wahrscheinlichkeitsrisiken durch eine dreifache Heterogenität:
  - materiell (unterschiedliche PCs, unterschiedliche BIOS, CEM-Kompatibilität gemäß der CEI1004 Norm mit Level 4)
  - softwarebezogen (zwei Rechner benutzen unterschiedliche C-Compiler)
  - zeitlich (Synchronität von Master und Slave, Ineinandergreifen der Anwendungsverarbeitung und der Selbsttests…)

- garantir la disponibilité du poste en minimisant les interventions de maintenance au moyen de :
  - un mécanisme de réinitialisation automatique extérieur (actif une fois) ;
  - une redondance en anneau des liens de communication sans câbles supplémentaires et commutation automatique des liaisons ;
  - des interfaces opérateurs de maintenance pour une vision «boîte blanche» et un archivage d'exécution non ambiguë correspondant strictement à la spécification ;
  - une interface claire entre matériel & logiciel fonctionnel afin d'autoriser la modification ou la régénération d'une partie sans être dans l'obligation de retoucher l'autre et réciproquement ;
- Faciliter l'exploitation, notamment en modes dégradés en minimisant les conséquences au moyen de modules d'enclenchements locaux (réduction des modes communs) avec possibilité de commande en local, d'effacement d'une gare ;
- Optimiser les coûts sur le cycle de vie du poste d'aiguillage :
  - la démonstration de sécurité de l'ensemble matériel et logiciel est indépendante du matériel (gestion des obsolescences, multi sources…) ;
  - l'utilisation du matériel industriel standard non développé spécifique pour le ferroviaire (PC…) ;
  - une limitation des travaux de non régression et de minimisation des essais avant mise en exploitation : une évolution du logiciel applicatif ne modifie pas le logiciel critique, une évolution du logiciel critique ne modifie pas logiciel applicatif.

Un atelier d'élaboration des graphes fonctionnels permet de générer rapidement avec un haut niveau de confiance, en cohérence avec une bibliothèque exhaustive, l'ensemble des graphes utiles pour les postes d'aiguillage. Seules les erreurs de compréhension des besoins métiers réels en amont de cet outil subsistent.

Cette orientation[57] a été reprise depuis dans la norme EN50128. [EN50128, 2001]

- Gewährleistung der Stellwerksverfügbarkeit durch Minimierung der Instandhaltungseingriffe mit :
  - einem automatischen Neustart (einmal aktiv),
  - einer ringförmigen Redundanz der Kommunikationsverbindungen, ohne zusätzliche Kabel und mit automatischer Verbindungsumschaltung
  - „White box" Schnittstellen für das Wartungspersonal und eine unmissverständliche Archivierung der Ausführungen, die strikt den Spezifikationen entsprechen
  - eine klare Schnittstelle zwischen Hardware und funktioneller Software, damit ein Teil unabhängig von den anderen geändert oder ersetzt werden kann.
- Erleichterung des Betriebes insbesondere in der Rückfallebene, durch Minimierung der Folgen: örtliche Sicherungsmodule (Reduzierung des gemeinsamen Versagens) mit der Möglichkeit einer Steuerung vor Ort
- Optimierung der Lebenszykluskosten des Stellwerks:
  - hardwareunabhängige Sicherheitsbeweisführung der Hardware und Softwareeinheit (Management der Veralterung, mehrfache Quellen…)
  - Verwendung einer industriellen Standardhardware, und nicht eines speziell für den Bahnbereich entwickelten Materials (PC…)
  - Beschränkung des Arbeitsaufwands bezüglich der Nichtverschlechterung und Minimierung der Tests vor der Inbetriebnahme: eine Veränderung der Anwendungssoftware beeinflusst nicht die kritische Software und eine Weiterentwicklung der kritischen Software verändert nicht die Anwendungssoftware.

Ein Programm für die schnelle Erzeugung funktioneller Graphen mit einer hohen Zuverlässigkeit erlaubt es, in Übereinstimmung mit einer erschöpfenden Graphenbibliothek alle für das Stellwerk nützlichen Graphen zu erstellen. Nur Verständnisfehler bezüglich der fachspezifischen Bedürfnisse vor dem Einsatz dieses Hilfsprogramms bleiben bestehen.

Diese Ausrichtung[58] wurde auch in die endgültige Fassung der Norm EN50128 aufgenommen. [EN50128, 2001]

[57] Orientation innovante à l'époque de la conception du SYMEL (Mise en service en juin 1995) et du PIPC (1997)

[58] Zum Zeitpunkt der Konzeption des SYMEL (MS, Juni 1995) und des PIPC (1997) war dies eine innovative Ausrichtung; in der genannten Norm wird sie weniger stark befürwortet.

## 6.2.2 Architecture logicielle de base du module d'enclenchement (MEI)

### 6.2.2.1 Moteur de résolution des graphes

Le moteur de résolution des graphes assure l'exécution dynamique des graphes fonctionnels. Le moteur de résolution des graphes comprend :

- la gestion des événements externes[59] (CTL changement d'état des entrées terrain, MSG réception d'un message externe, FTP échéance d'une temporisation, ACT activation d'un graphe par un autre) ;
- la gestion des événements internes[60] résultant du franchissement des transitions (IND positionnement d'une variable interne, ACT activation d'un automate) ;
- le traitement d'un événement (interne ou externe) ;
- la gestion des variables internes manipulées dans les graphes constituée :
  - la lecture de l'état courant d'une variable interne ;
  - le positionnement d'une variable interne dans un état désiré ;
  - la détection des changements d'état des variables internes ;
- l'activation des processus chargés du traitement des actions entreprises lors du franchissement d'une transition :
  - l'activation d'une variable interne (IND : injection d'un événement interne dans le moteur) ;
  - l'activation d'un automate (ACT: injection d'un événement interne dans le moteur) ;
  - l'activation d'une sortie terrain (CMD) ;
  - l'armement ou arrêt d'une temporisation (DTP ou ATP).

L'activation d'une variable interne est différée jusqu'à la fin du traitement de l'événement courant, c'est à dire lorsque toutes les transitions potentielles associées à l'événement traité ont été examinées avec le même contexte. L'action est donc d'abord mémorisée dans une file d'attente (FIFO) des actions différées avant d'être effectivement réalisée. Si un changement d'état a été détecté, injection l'événement interne correspondant.

[59] Evénements : CTL entrée terrain, FTP échéance d'une temporisation, DTP début de temporisation, ATP arrêt de temporisation, MSG réception d'un message / Actions : MSG émission d'un message, CMD sortie terrain

[60] Evénements : IND variable interne résumant l'état d'un graphe, nommé indicateur, ACT action interne inter-graphes

## 6.2.2 Grundsoftwarearchitektur des Sicherungsmoduls (MEI)

### 6.2.2.1 Graphenlösungsmaschine

Die Maschine zur Lösung der Graphen sichert die dynamische Ausführung der Funktionen. Die Graphenlösungsmaschine umfasst:

- die Verwaltung externer Ereignisse[61] (CTL Zustandsänderung der Eingänge vor Ort, MSG Empfang einer externen Meldung, FTP Ende einer Verzögerung, ACT Aktivierung eines Graphen durch einen anderen),
- die Verwaltung interner Ereignisse[62], die aus der Überschreitung von Übergängen herrühren (IND Stellung einer internen Variable, ACT Aktivierung eines Graphen),
- die Bearbeitung eines (internen oder externen) Ereignisses,
- die Verwaltung der in den Graphen benutzten internen Variablen:
  - das Einlesen des laufenden Zustands einer internen Variable,
  - Überführung einer internen Variablen in einen gewünschten Zustand,
  - Feststellung der Zustandsänderungen der internen Variablen,
- die Aktivierung der Prozesse zur Bearbeitung der bei der Überschreitung eines Übergangs eingeleiteten Aktionen:
  - die Aktivierung einer internen Variable (IND: Einspeisung eines internen Ereignisses in die Maschine),
  - die Aktivierung eines Graphen (ACT: Einspeisung eines internen Ereignisses in die Maschine),
  - die Aktivierung einer Ausgabe vor Ort (CMD),
  - das Beenden einer Verzögerung (DTP oder ATP).

Die Aktivierung einer internen Variablen wird bis zum Ende der Bearbeitung des laufenden Ereignisses aufgeschoben, d. h. bis alle zu dem bearbeitenden Ereignis gehörenden potentiellen Übergänge in demselben Zusammenhang geprüft wurden. Die Handlung wird also zuerst in einer Warteschlange (FIFO) der aufgeschobenen Aktionen gespeichert, bevor sie tatsächlich ausgeführt wird. Wird eine Zustandsänderung festgestellt, so wird das entsprechende interne Ereignis eingespeist.

[61] Ereignis : CTL Eingang vor Ort, FTP Ende einer Verzögerung, DTP Anfang einer Verzögerung, ATP Stoppen einer Verzögerung, MSG Empfang einer Meldung / Handlung : MSG Senden einer Meldung, CMD Ausgang vor Ort.

[62] Ereignis : IND interne Variable, die den Zustand eines Graphen zusammenfasst und als Indikator bezeichnet wird, ACT Handlungen zwischen Graphen

L'initialisation dynamique des graphes : initialisation du marquage de chacun des graphes est réalisée par injection à la mise sous tension d'un événement d'initialisation pour chaque automate utilisé (transition 0 vers X).

Le moteur de résolution est purement événementiel, c'est à dire qu'une transition ne peut être franchie que sur présentation d'un événement.

**Remarque :**

Un trait commun des systèmes critiques ferroviaires est la nécessité de concevoir chaque installation en vue de répondre aux exigences individuelles d'une application spécifique.

La mise en place d'un système configuré par les données permet la mise en place d'un logiciel générique[63] : dans notre cas nous allons plus loin, le logiciel de base est non seulement générique (recompilé pour chaque site) mais identique à l'ensemble des sites.

Cette option technique permet une séparation code/donnée. Le processus de développement mis en place pour la réalisation du logiciel de base aurait pu se faire en utilisant une méthode formelle (langage B par exemple). La présence d'une preuve formelle de conformité de code développé aux spécifications permet de réduire encore les phases de test du cycle en V, mais elle ne règle pas le problème dans le cas des données de sécurité.

Le fonctionnel applicatif instancié entre dans le champ des données de sécurité et fait l'objet de notre validation méthode.

Die dynamische Initialisierung der Graphen: Initialisierung der Markierung jedes Graphen durch Einspeisung eines Initialisierungsereignisses für jeden verwendeten Graphen (Übergang Platz 0 nach X).

Die Lösungsmaschine ist rein ereignisbezogen, d. h. ein Übergang ist nur bei Vorhandensein eines Ereignisses möglich.

**Bemerkung:**

Eine gemeinsame Charakteristik kritischer Bahnsysteme ist die Notwendigkeit, jede Anlage so zu entwerfen, dass sie den individuellen Anforderungen einer spezifischen Anwendung entspricht.

Die Schaffung eines durch Daten konfigurierten Systems erlaubt es, eine generische[64] Software zu schaffen. In dem vorgestellten Fall wird noch ein Schritt weiter gegangen: die Grundsoftware ist nicht nur generisch (und für jeden Standort neu kompiliert), sondern sie ist für alle Standorte identisch.

Diese Wahl erlaubt es, den Code von den Daten zu trennen. Das Entwicklungsverfahren hätte auf ein formales Verfahren (z. B. B-Sprache) zurückgreifen können. Die Durchführung eines formalen Konformitätsbeweises zwischen dem entwickelten Code und den Spezifikationen erlaubt es, die Testphasen des V-Zyklus zu reduzieren, aber sie löst das Problem der Sicherheitsdaten nicht.

Die anwendungsbezogenen, parametrisierten Funktionen gehören zu den Sicherheitsdaten und sind Gegenstand der in dieser Arbeit vorgestellten formalen Validierung.

[63] Conforme à la norme européenne NF EN50128 « Application ferroviaires – logiciels pour systèmes de commande et de protection ferroviaire », Juillet 2001 [EN50128, 2001]

[64] Entsprechend der Norm EN50128 : „Bahnanwendungen - Telekommunikationstechnik, Signaltechnik und Datenverarbeitungssysteme - Software für Eisenbahnsteuerungs- und Überwachungssysteme“, Juli 2001 [EN50128, 2001]

### 6.2.2.2 Procédures d'injection d'événements dans les graphes

Pour chacun des événements externes injectés, il est procédé à l'activation des graphes pour la prise en compte de l'événement. L'ordre d'évaluation des événements est le suivant :

- traitement d'un événement externe (premier événement mémorisé) ;
- traitement de tous les événements internes résultant de la prise en compte du dernier événement externe, des événements internes générés durant le traitement ;
- traitement de l'événement externe suivant, etc.

Pour cela, il existe deux files d'attente d'événements :

- la file d'attente des événements externes (changement d'état d'une entrée terrain, échéance de temporisation, activation de variables internes) ;
- la file d'attente des événements internes (changement d'état d'une variable interne, activation d'un graphe).

Lors de la prise en compte d'un événement par le fonctionnel applicatif (écrite sous forme de réseaux de Petri cf. Figures 6.7 à 6.14), les traitements effectués sont :

- la recherche parmi l'ensemble des graphes des transitions potentiellement concernées par cet événement, c'est-à-dire toutes les transitions pour lesquelles cet événement est potentiellement initiateur ;
- la vérification de leur marquage sur la place origine ;
- la recherche parmi ces transitions des transitions franchissables ;
- l'évaluation des conditions de franchissement pour chacune de ces transitions franchissables (ou s'il n'y a pas de condition) :
  - activation des traitements applicatifs associés au franchissement de la transition (actions entreprises depuis les graphes) ;
  - actualisation du marquage dû au franchissement de la transition ;
  - sinon, activation des traitements applicatifs associés au non franchissement de la transition marquée (traitement des alarmes).

### 6.2.2.2 Verfahren zur Einspeisung von Ereignissen in die Graphen

Für jedes der externen eingespeisten Ereignisse erfolgt eine Aktivierung der Graphen zur Berücksichtigung des Ereignisses. Die Reihenfolge der Auswertung der verschiedenen Ereignisse ist folgende:

- Bearbeitung eines externen Ereignisses (erstes gespeichertes Ereignis)
- Bearbeitung aller interner Ereignisse unter Berücksichtigung des letzten externen Ereignisses, das während der Verarbeitung erzeugt wurde
- Bearbeitung des folgenden externen Ereignisses, usw.

Hierfür gibt es zwei Warteschlangen von Ereignissen:

- die Warteschlange der externen Ereignisse (Zustandsänderung eines Eingangsparameters vor Ort, Ende der Verzögerungszeit, Aktivierung von internen Variablen durch die anwendungsbezogene Software)
- die Warteschlange der internen Ereignisse (Zustandsänderung einer internen Variablen, Aktivierung eines Graphen).

Bei der Berücksichtigung eines Ereignisses durch die anwendungsbezogenen Funktionen (durch Petrinetze beschrieben Abb. 6.7 bis 6.14) wird folgendes durchgeführt:

- die Suche innerhalb der Graphen der von diesem Ereignis potentiell betroffenen Transitionen, d.h. aller Transitionen, die potentiell von diesem Ereignis geschaltet wurden
- die Prüfung ihrer Markierung (Abb. 6.4) auf dem Ursprungsplatz
- die Suche innerhalb dieser Transitionen nach feuerbaren Transitionen
- die Bewertung der Bedingungen zum Fortschalten jeder dieser Transitionen (oder Feststellung, dass es keine Bedingung gibt)
  - Aktivierung der anwendungsbezogenen Bearbeitungen in Verbindung mit dem Fortschalten
  - Aktivierung der Markierung nach dem Fortschalten
  - ansonsten Aktivierung der anwendungsbezogenen Bearbeitung bezüglich des Nichtschaltens der markierten feuerbaren Transitionen (Alarmbehandlung).

L'évaluation de la condition d'une transition correspond au booléen VRAI si les conditions de franchissement sont vérifiées. Elle permet alors de donner à la transition potentielle en cours d'évaluation vis-à-vis de l'événement traité le statut de transition franchissable.

Les conditions dans les graphes sont exprimées exclusivement à l'aide de produits de sommes ou de sommes de produits de variables booléennes.

Cette manière dont l'interpréteur traite les informations permet d'obtenir un fonctionnement du déterministe du système, notamment quant à la façon et dans l'ordre dans lequel les informations sont traitées.

Remarques :

L'ordre dans lequel les actions sont réalisées dépend de l'ordre dans lequel les transitions des automates sont franchies. Or cet ordre est important car le réseau d'automates peut avoir un comportement différent en fonction de cet ordre. En fait, dans une machine, l'interpréteur traite les automates dans un ordre bien déterminé défini au moment de l'implantation du réseau d'automates.

Dans notre cas, il est à noter que cet ordre de franchissement n'a aucun effet sur le fonctionnement de l'automate.

Aussi, une des contraintes de conception de ces graphes est que l'ordre d'évaluation des automates afin de trouver les transitions potentielles à la survenue d'un événement n'est pas connu a priori lors de l'écriture des graphes génériques !

(voir l'exemple du paragraphe 6.2.4.9)

Die Bewertung der Bedingung für eine Transition entspricht dem booleschen Operator WAHR, wenn die Bedingungen zum Schalten überprüft sind. Die im Hinblick auf das bearbeitete Ereignis bewertete potentielle Transition bekommt somit den Status einer feuerbaren Transition.

Die Bedingungen innerhalb der Graphen werden ausschließlich mit Hilfe von Produktsummen oder von Summen von Produkten boolescher Operatoren ausgedrückt.

Die Art und Weise, in welcher der Interpreter die Informationen bearbeitet, erlaubt es, eine deterministische Funktionsweise des Systems hinsichtlich der Funktionsweise und der Reihenfolge der Informationsverarbeitung zu erhalten.

Bemerkung:

Die Reihenfolge, in der die Aktionen ausgeführt werden, hängt von der Reihenfolge ab, in der die Transitionen der Automaten geschaltet werden. Diese Reihenfolge ist wichtig, denn das Automatennetz kann sich je nach Reihenfolge unterschiedlich verhalten. In einer Maschine bearbeitet der Interpreter die Automaten in einer ganz bestimmten Reihenfolge; diese wird zum Zeitpunk der Implementierung des Automatennetzes bestimmt.

Für den hier vorgestellten Fall ist ferner anzumerken, dass die Reihenfolge der Schaltungen die Funktionsweise des Automaten nicht beeinflusst.

Eine der randbedingungen beim Entwurf dieser Graphen ist somit die Tatsache, dass die Reihenfolge der Auswertung der Automaten, die beim Eintreten eines Ereignisses alle möglichen Transitionen finden soll, a priori nicht zum Zeitpunkt des Schreibens der allegemeingültigen Graphen bekannt ist.

(Siehe Beispiel am Paragraph 6.2.4.9)

### 6.2.2.3 Gestion des temporisations par le logiciel de base

Les temporisations permettent de retarder l'exécution d'une procédure d'un intervalle de temps déterminé. Cette gestion des temporisations relève de la gestion dans le temps d'un traitement au départ différé. La gestion des temporisations par le logiciel de base consiste à mettre à disposition du fonctionnel :

- Une commande d'armement d'une temporisation (DTP) ;
- Une commande d'arrêt d'une temporisation en cours (ATP) ;
- Un événement survenant lorsque la temporisation est échue (FTP).

### 6.2.2.4 Conséquences pour la preuve

Le module d'enclenchement (MEI) comporte des propriétés fondamentales de fonctionnement (strictement déterministe) pour appliquer la méthode proposée :

- il conserve la chronologie de tous les évènements externes, quelle que soit l'espacement entre ces évènements. Tous ces évènements seront pris en compte ensuite dans l'ordre de leur apparition ;
- il traite un seul évènement externe à la fois. Quand un évènement externe est pris en compte, cet évènement est «propagé» dans l'ensemble des graphes qui constituent le fonctionnel applicatif. Une fois la propagation interne terminée, l'évènement externe suivant est alors pris en compte ;
- il interprète les graphes tels quels, sans aucune réécriture, ni programmation algorithmique ;
- il comporte des règles d'interprétation fixées, les équations de transitions, avec leur grammaire et leur vocabulaire, sont lus dans un ordre défini ;
- il se comporte comme une machine abstraite à temps de commutation nulle (toutes les transitions sont instantanées).

### 6.2.2.3 Verwaltung der Verzögerungen durch die Grundsoftware

Die Verzögerungszeiten verschieben die Ausführung eines Ablaufs um eine bestimmte Zeit. Die Verwaltung der Verzögerungen fällt in den Bereich der Zeitverwaltung verzögerter Bearbeitungen. Bei der Verwaltung der Verzögerungszeiten durch die Grundsoftware wird den Funktionen Folgendes zur Verfügung gestellt:

- Aktivieren der Verzögerungszeit (DTP)
- Deaktivieren einer zuvor aktivierten Verzögerungszeit (ATP)
- Wecken eines Ereignisses nach Ablauf der Verzögerungszeit (FTP).

### 6.2.2.4 Folgen für den Beweis

Das Sicherungsmodul (MEI) beinhaltet grundlegende (strikt deterministische) Funktionseigenschaften zur Anwendung der vorgeschlagenen Methode:

- Es behält die Reihenfolge aller externen Ereignisse, unabhängig von der Zeit zwischen den Ereignissen, bei. Alle diese Ereignisse werden danach in der Reihenfolge ihres Auftretens bearbeitet.
- Es bearbeitet jeweils nur ein externes „Ereignis“. Es wird ein externes „Ereignis“ bearbeitet, so wird es auf alle Graphen der anwendungsbezogenen Funktionseinheit „verteilt“. Nach der internen Verteilung wird das nächste externe Ereignis bearbeitet.
- Es interpretiert die Graphen so wie sie sind, ohne Umschreibung und ohne algorithmische Programmierung.
- Es umfasst feste Interpretationsregeln, Transitionsgleichungen mit Grammatik und Vokabular; diese werden in einer definierten Reihenfolge eingelesen.
- Es verhält sich wie eine abstrakte Maschine ohne Kommunikationszeit (alle Schaltungen erfolgen sofort).

La figure 6.5 clarifie les particularités d'interprétation des graphes réalisant le fonctionnel applicatif du poste.

Abb. 6.5 versucht, die Besonderheiten bei der Interpretation der Graphen, die die anwendungsbezogene Funktionseinheit des Stellwerkes darstellen, zu erklären.

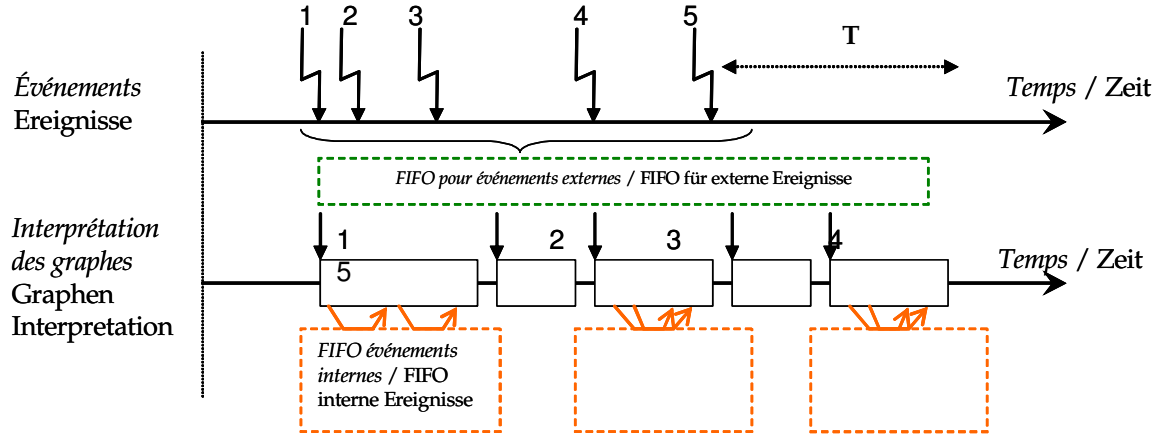



**Figure 6.5 :** Gestion des entrées: fonctionnement d'un automate à temps de transition nul
**Abbildung 6.5:** Verwaltung der Eingänge: Funktionsweise eines Automaten ohne Übergangszeit

L'utilisation de réseaux de Petri interprétés (réseaux particuliers que nous allons décrire plus précisément dans le §6.2.3) permet d'éviter tout recours à une écriture algorithmique. Cette option permet :

- d'avoir une écriture et une lecture abordable des fonctionnalités métier ;
- de confondre le langage formel de description et le langage interprété/exécuté par la machine cible.

Commentaire :
Il est à noter que les règles d'interprétation sont différentes de celles des pour les réseaux SIPN sur plusieurs points:

- le traitement différentié des événements externes et internes ;
- la gestion des indicateurs
- absence de hiérarchie des graphes ;
- la propagation des événements internes ;
- la gestion des temporisations.

Les réseaux de Petri interprétés SIPN [Frey, 1998] [Frey, 1999] [Frey, 2002] [Frey, 2003] [Tarnai, 2009] constituent néanmoins avec les Grafcet l'écriture de réseaux de Petri la plus proche de celle décrite dans ce travail.
Cette approche a fait l'objet d'un livre de König et Quack. Frey s'appuie sur ces travaux.

Die Verwendung der interpretierten Petrinetze (besondere Netze, die unter §6.2.3 näher beschrieben werden) erlaubt es, auf das algorithmische Schreiben zu verzichten. Diese Wahl erlaubt es:

- das Lesen und Schreiben der anwendungsbezogenen fachspezifischen Funktionsweisen zu erleichtern.
- die formale Sprache der Beschreibung und die interpretierte/ausgeführte Sprache der Zielmaschine zu vereinen.

Kommentar:
Es ist anzumerken, dass die Interpretationsregeln für die SIPN Netze in mehreren Punkten verschieden sind:

- die unterschiedliche Verarbeitung von externen und internen Ereignissen
- die Verwaltung der Indikatoren
- keine Hierarchie der Graphen
- die Ausbreitung von internen Ereignisse
- die Verwaltung von Verzögerungen

Die interpretierten Petrinetze SIPN [Frey, 1998] [Frey, 1999] [Frey, 2002] [Frey, 2003] [Tarnai, 2009] kommen, zusammen mit den Grafcet, den in dieser Arbeit dargestellten Petrinetzen am nächsten. Die SIPN kommen den in dieser Arbeit beschriebenen Petrinetzen an näherten.
Diese werden von König und Quack in einem Buch vorgestellt. Frey bezieht sich darauf.

## 6.2.3 Logiciel fonctionnel applicatif du module d'enclenchement (MEI)

### 6.2.3.1 Langage interprété et langage de description de l'applicatif

Les spécifications fonctionnelles des postes d'aiguillages reposent sur les réseaux de Petri. Des graphes d'états suffisent pour des processus séquentiels et combinatoires que sont les fonctionnels applicatifs des postes d'aiguillage. Comme nous le verrons plus loin, une des caractéristiques de la méthode présentée est de valider formellement l'applicatif qui sera implanté dans la machine cible en utilisant les propriétés des réseaux de Pétri. Il n'y a donc aucune différence entre le modèle de haut niveau et le code exécuté.

Le fonctionnel applicatif du poste est décrit sous forme d'un réseau d'automates à états finis. Ces graphes décrivent le fonctionnement des enclenchements de signalisation. Chaque graphe ne peut se trouver que dans un nombre fini d'états et ne peut être que dans un seul état à un instant donné. Chaque état d'un graphe est représenté par *une et une seule* place.

Les états d'un même graphe sont reliés par des transitions. Une transition est composée d'un champ événement, d'un champ condition et d'un champ action. Les éléments définis dans les transitions sont appelées entités fonctionnelles et servent à l'identification des éléments physiques extérieurs au système et des variables internes. Chaque entité fonctionnelle est une variable *booléenne* dont le type est déterminé par un préfixe d'identification :

| Préfixe | Type d'entité fonctionnelle |
|---|---|
| CTL | Contrôle d'entrée terrain (Lecture de l'état d'un relais ou d'un commutateur) |
| CMD | Commande de sortie terrain (Cde de relais) |
| ACT | Activateur d'automate (Sert à la synchronisation entre automates) |
| IND | Indicateurs (Variable interne) |
| MSG | Message (commande envoyée par un agent) |
| FCI | Fonction de commande informatique (Appel de cette fonction) |
| DTP | Début de temporisation |
| ATP | Arrêt de temporisation |
| FTP | Fin de temporisation |

**Tableau 6.1** : Type de variables logiques pouvant être gérées par l'interpréteur de graphe

## 6.2.3 Funktionelle anwendungsbezogene Software des Sicherungsmoduls (MEI)

### 6.2.3.1 Interpretierte Sprache und Beschreibung der Anwendung

Die funktionale Spezifikation des Stellwerks beruht auf Petrinetzen, die aufgrund ihrer mathematischen Eigenschaften benützt werden. In der Tat genügen einfache Zustandsgraphen, um die sequentiellen Kombinationsvorgänge der anwendungsbezogenen Funktionseinheiten des Stellwerks zu beschreiben. Wie später gezeigt wird, besteht eine der Eigenschaften des vorgestellten Verfahrens in der formalen Validierung der Anwendungen der Zielmaschine, unter Verwendung der Eigenschaften der Petrinetze. Es gibt also keinen Unterschied zwischen dem Modell auf hoher Ebene und dem ausgeführten Code.

Die anwendungsbezogenen Funktionen des Stellwerks werden durch ein Netz von endlichen Automaten beschrieben. Diese Graphen beschreiben die Funktionsweise der Signalsicherung. Jeder Graph kann sich nur in einer endlichen Anzahl von Zuständen befinden, und zu einem gegebenen Zeitpunkt kann er sich nur in einem einzigen Zustand befinden. Jeder Zustand eines Graphen wird durch *einen einzigen Platz* dargestellt.

Die Zustände eines Graphs sind durch die Übergänge miteinander verbunden. Ein Eingang besteht aus einem Feld „Ereignis", einem Feld „Bedingung" und einem Feld „Handlung". Die Elemente in den Schaltungen heißen „funktionale Einheiten". Sie dienen der Identifizierung der externen physischen Elemente des Systems und der internen Variablen. Jede funktionale Einheit ist eine boolesche Variable, deren Typ von einem Identifizierungspräfix bestimmt wird (siehe Tab. 6.1)

| Präfix | Art der funktionellen Einheit |
|---|---|
| CTL | Eingangskontrolle vor Ort (Lesen des Zustandes eines Relais oder eines Schalters) |
| CMD | Ausgangsanssteuerung vor Ort (Relaisansteuerung) |
| ACT | Automatenaktivator (zur Synchronisierung zwischen den Automaten) |
| IND | Indikatoren (interne Variable) |
| MSG | Meldung (Kommunikation durch einen Bediensteten) |
| FCI | IT-Steuerfunktion (Aufrufen dieser Funktion) |
| DTP | Beginn der Verzögerungszeit |
| ATP | Unterbrechung der Verzögerungszeit |
| FTP | Ende der Verzögerungszeit |

**Tabelle 6.1** : Arten der logischen Variablen, die von dem Graphinterpreter verwaltet werden können

Sur une place donnée un indicateur prend toujours la même valeur. Un indicateur du fonctionnel peut être modifié que par un seul automate mais l'ensemble des automates peut accéder à sa valeur ou le faire entrer en ligne de compte dans une transition (événement, condition). Le tableau 6.2 récapitule les possibilités d'utilisation :

| Préfixe | Événement | Condition | Action |
|---|---|---|---|
| CTL | X (si changement d'état) | X | |
| CMD | | | X |
| ACT | X | | X |
| IND | X (si changement d'état) | X | X |
| MSG | X | | |
| FCI | | | X |
| DTP | | | X |
| ATP | | | X |
| FTP | X | | |

**Tableau 6.2** : Utilisation des variables logiques gérées par l'interpréteur de graphe

Les fonctionnalités (principes de signalisation) d'un poste d'aiguillage sont un ensemble de graphes communiquant entre eux. Ils sont en interaction avec leur environnement. Il s'agit précisément de :

- des machines à états. Ce sont des réseaux de Petri tel que chaque transition soit reliée à exactement une place en entrée et une place en sortie par des arcs de valeur 1 ;
- des réseaux non autonomes. C'est une classe des réseaux de Petri dont l'évolution ne dépend pas uniquement de l'état du réseau mais aussi de l'état de l'environnement associé ;
- des réseaux sont impurs. En effet, certaines places peuvent être à la fois entrée et sortie d'une même transition ;
- des réseaux de Petri ont un marquage binaire.

Le changement d'état d'un graphe est toujours lié à un changement externe.

Auf einem bestimmten Platz nimmt der Indikator immer denselben Wert an. Der Indikator einer Funktion kann nur durch einen einzigen Automaten geändert werden, aber alle Automaten haben Zugang zu seinem Wert oder können ihn in einer Transition (Ereignis, Bedingung) berücksichtigen. Tabelle 6.2 beinhaltet die möglichen Verwendungen.

| Präfix | Ereignis | Bedingung | Aktion |
|---|---|---|---|
| CTL | X (im Falle einer Zustandsänderung) | X | |
| CMD | | | X |
| ACT | X | | X |
| IND | X (im Falle einer Zustandsänderung) | X | X |
| MSG | X | | |
| FCI | | | X |
| DTP | | | X |
| ATP | | | X |
| FTP | X | | |

**Tabelle 6.2**: Verwendung der vom Graphinterpreter erzeugten Variablen

Die Funktionen (Signalgebungsprinzipien) eines Stellwerkes sind eine Reihe von Graphen, die miteinander kommunizieren. Es gibt eine Wechselwirkung zwischen ihnen und ihrem Umfeld. Es handelt sich insbesondere um Folgendes:

- Zustandsmaschinen: Petrinetze, bei denen jede Transition exakt in Verbindung steht mit einem Eingangsplatz und einem Ausgangsplatz durch eine Kante mit dem Wert 1 verbunden. Die Verzweige- und Begrenzungssuche ist für alle Funktionen identisch.
- Nicht autonome Netze. Dabei handelt es sich um eine Klasse von Petrinetzen, deren Entwicklung nicht nur vom Netzzustand, sondern auch von dem betreffenden Zustand des Umfeldes abhängt.
- Bestimmte Netze sind „unrein": bestimmte Plätze sind sowohl Eingang als auch Ausgang ein und derselben Transition. (Schlingen)
- Petrinetze sind binär markiert.

Die Zustandsänderung eines Graphen hängt immer mit einem externen „Ereignis" zusammen.

### 6.2.3.2 Limite des outils classiquement utilisés pour interpréter les RdP

Une attention particulière doit être portée sur les possibilités et limitations des interpréteurs de réseaux de Petri disponibles. Les règles habituelles d'interprétation conduisent rapidement à une «explosion combinatoire» et/ou à un «non déterminisme d'exécution» si l'on n'y prend garde. Le choix des réseaux de Pétri pour ses propriétés mathématiques n'est pas à remettre en cause, il est néanmoins nécessaire d'identifier les faiblesses des interpréteurs[65] (cf. §6.2.2.2) :

- la non distinction d'événements externes et internes, supposant ainsi de fait que tout événement externe ne peut pas engendrer de transitions en cascade (l'événement active le graphe A qui lui-même active le graphe B et ainsi de suite). L'exploration met alors sur un pied d'égalité un nouvel événement externe et l'événement interne entre les graphes A et B, créant ainsi des états systèmes fictifs, sans réalisé physique. Or une des forces des réseaux de Petri consiste justement à représenter le fonctionnel applicatif par des graphes simples, compréhensibles par les experts signalisation. Notons que le problème disparaît si l'on interprète un graphe produit de l'ensemble des graphes élémentaires, perdant ainsi l'avantage de la modularité des graphes élémentaires. Il y a donc nécessité de distinguer dans l'interprétation et l'exploration les événements internes et les événements externes de l'ensemble de graphes ;
- la dépendance par rapport à l'ordre d'écriture des graphes. Le franchissement d'une transition entraîne de suite la mise à jour du marquage, cette mise à jour peut alors modifier les conditions de franchissement (en + ou en -) d'autres transitions qui seront évaluées ultérieurement (graphe de rang plus élevé). L'inversion de l'ordre d'interprétation des graphes, la modification des priorités d'évaluation, ou l'insertion d'un nouveau graphe peut modifier le fonctionnement de l'automate (ensemble des graphes). Cf. Annexe D qui présente un exemple.

[65] Logiciels ROMEO de l'Ecole Centrale de Nantes ou PNEDIT de l'université de Braunschweig par exemple

### 6.2.3.2 Grenzen der klassischen PN-Interpreter

Eine besondere Beachtung muss den Möglichkeiten und den Grenzen der Petrinetzinterpreter geschenkt werden. Die üblichen Interpretationsregeln führen, wenn man nicht darauf achtet, schnell zu einer „kombinatorischen Explosion" und/oder zu „nichtdeterministischen Ausführungen". Die Wahl der Petrinetze aufgrund ihrer mathematischen Eigenschaften wird nicht in Frage gestellt. Es ist trotzdem notwendig, die Schwächen der Interpreter zu identifizieren[66] (siehe §6.2.2.2):

- keine Unterscheidung zwischen externen und internen Ereignissen. Die Interpreter gehen davon aus, dass kein externes Ereignis eine Kaskade von Schaltungen verursachen kann (das Ereignis aktiviert den Graphen A, der selbst den Graphen B aktiviert etc.). Die Auswertung behandelt so ein neues externes Ereignis und ein internes Ereignis zwischen den Graphen A und B auf die gleiche Weise. Das lässt fiktive Systemzustände entstehen, die keinen physischen Gegenpart haben. Eine Stärke der Petrinetze besteht aber genau darin, die funktionelle Anwendung durch einfache, für die Signaltechnikexperten leicht verständliche Graphen, darzustellen. Es ist anzumerken, dass dieses Problem verschwindet, wenn man den Produktgraphen aller elementaren Graphen interpretiert. Man verliert dann allerdings den Vorteil der Modularität der elementaren Graphen. Es ist also notwendig, bei der Interpretation und der Auswertung die internen und die externen Ereignisse für alle Graphen zu unterscheiden
- die Abhängigkeit hinsichtlich der Reihenfolge des Schreibens der Graphen. Die Schaltung einer Transition bewirkt gleichzeitig die Aktualisierung der Markierung. Diese Aktualisierung kann dann die Bedingungen der Schaltung anderer Übergänge die später ausgewertet werden (nach der Rangordnung höhere Graphen) ändern (positiver oder negativer Einfluss). Die Veränderung der Reihenfolge der Interpretation der Graphen, die Änderung des Vorrangs bei der Auswertung oder das Einfügen eines neuen Graphen kann das Funktionieren des Automaten (Gesamtheit der Graphen) beeinflussen.

[66] Die von der Ecole Centrale de Nantes entwickelte Software ROMEO oder die von der TUBS entwickelte Software PNEDIT zum Beispiel

Il est donc nécessaire de distinguer le marquage futur des places et le marquage courant des places pour l'évaluation des conditions booléennes des transitions des graphes. C'est pourquoi nous avons introduit en plus du marquage des places des « indicateurs » pour l'évaluation des champs conditions ;

- l'écriture des propriétés de sécurité sous la forme de fonctions combinatoires. L'écriture des obligations de preuve dans un langage formel s'opère au moyen d'invariants à vérifier quel que soit l'état fonctionnel courant du système à valider. L'écriture de ces obligations de preuve nécessite une très bonne connaissance du langage formel afin d'utiliser divers «trucs et astuces» pour contourner cette difficulté. Ce niveau d'expertise n'est pas accessible aux experts signalisation.
  Or ce sont les experts les mieux qualifiés pour exprimer les propriétés de sécurité !
  C'est pourquoi nous avons utilisé pour l'écriture des propriétés de sécurité un langage formel accessible aux experts signalisation, en l'occurrence le même langage que celui utilisé pour le logiciel fonctionnel applicatif[67].

C'est pour ces raisons que nous avons définis le langage AEFD (Automate à États Finis déterministes) que nous allons présenter au plus avant.

Es ist also notwendig, die künftige und die laufende Markierung der Zustände bei der Bewertung der booleschen Bedingungen der Transitionen der Graphen zu unterscheiden. Deshalb wurden zusätzlich zu der Markierung der Zustände „Indikatoren" für die Bewertung der Bedingungsfelder eingeführt.

- das Schreiben der Sicherheitseigenschaften in Form von kombinatorischen Funktionen. Das Schreiben der Beweispflichten in einer formalen Sprache findet mithilfe von Invarianten statt, ungeachtet des laufenden funktionellen Zustands des zu prüfenden Systems. Das Schreiben dieser Beweispflichten erfordert eine sehr gute Kenntnis der formalen Sprache, um verschiedene „Tricks" zu benutzen, die diese Schwierigkeiten umgehen. Dieses Wissen ist bei den Signaltechnikexperten nicht vorhanden, obwohl nur die bestqualifizierten Fachleute die Sicherheitseigenschaften formulieren. Deshalb wurde für das Formulieren der Sicherheitseigenschaften eine formale Sprache benutzt, die den Signaltechnikexperten zugänglich ist. Im vorliegenden Fall handelt es sich um dieselbe Sprache wie die für die anwendungsspezifische funktionelle Software[68].

Aus diesem Grund wurde die AEFD-Sprache definiert (ein endlicher Automat), die nun im Detail vorgestellt wird.

[67] Le langage AEFD est utilisé avec succès depuis 1995 pour l'écriture du logiciel fonctionnel applicatif de SYMEL et poste PIPC.

[68] Die AEFD-Sprache wird seit 1995 erfolgreich für die Formulierung von funktionaler anwendungsbezogener Software des SYMEL und des PIPC Stellwerks verwendet.

## 6.2.4 Langage de spécification AEFD

### 6.2.4.1 Le langage d'écriture du fonctionnel ou langage AEFD

Le langage de description du fonctionnel (logiciel d'application) est, comme nous l'avons vu, une écriture particulière des automates concurrentiels à contraintes (de la famille des réseaux de Petri) compatible :

- avec la réalisation d'un automate industriel déterministe à piles FIFO ;
- la réalisation de preuves mathématiques sur les automates définis pour la description du fonctionnel sous forme de multiples graphes en communication.

Le langage AEFD permet de donner concrètement aux réseaux ainsi constitués les propriétés suivantes :

- l'interprétation événementielle du réseau est déterministe quelle que soit l'ordre d'interprétation des graphes, avec pour contraintes qu'un seul événement ne soit traité à la fois, que les événements externes survenant simultanément soient traités de manière décidable et que les graphes ne soient pas temporisées ;
- l'automate ainsi défini doit se comporter comme un automate à nombre fini d'états ;
- l'automate ainsi défini doit être interprétable en temps réel par un logiciel d'accueil d'une machine cible.

Nous avons nommé notre langage de description du nom de AEFD (Automates Événements Finis Déterministes). Dans ce langage il n'est ainsi jamais nécessaire d'utiliser des « priorités » ou des « temporisations » entre transitions.

Le langage AEFD reprend le formalisme utilisé au paragraphe précédent qui diffère par rapport aux écritures classiques par :

- l'introduction d'une écriture des graphes sous forme d'un fichier texte dit à « 6 lignes » ;
- l'introduction de la notion d'Indicateur qui met à jour APRES l'épuisement des changements de marquage ;
- le temps n'intervient pas en tant que tel dans les graphes, seul un événement dit de « fin de temporisation » est utilisé ;
- l'introduction des notions d'événements externes et internes afin de garantir une interprétation déterministe.

## 6.2.4 Spezifikationssprache AEFD

### 6.2.4.1 Sprache für das Formulieren von Funktionen oder AEFD-Sprache

Die Formulierungssprache von Funktionen ist, wie schon gezeigt wurde, eine besondere Formulierung nebenläufiger Automaten mit Randbedingung (aus der Familie der Petrinetze) die kompatibel sind mit:

- der Verwirklichung eines deterministischen industriellen FIFO Stapelautomaten,
- der Verwirklichung mathematischer Beweise auf den Automaten, die für die Beschreibung der Funktionen in Form von mehrfachen kommunizierenden Graphen definiert sind.

Die AEFD-Sprache erlaubt, den so dargestellten Netzen die folgenden Eigenschaften konkret zu geben:

- die Ereignisinterpretation des Netzes ist deterministisch, ungeachtet der Reihenfolge der Graphinterpretation, mit der Einschränkung, dass nur ein Ereignis zur Zeit behandelt wird, dass die externen Ereignisse, die gleichzeitig auftreten, auf entscheidbare Art behandelt werden und dass die Graphen nicht verzögert sind,
- der so definierte Automat muss sich wie ein Automat mit einer endlichen Anzahl von Zuständen verhalten,
- der so definierte Automat muss in Echtzeit durch eine Software auf der Zielmaschine interpretierbar sein.

Diese Formulierungssprache wurde AEFD genannt (Automaten mit Deterministischen und Endlichen Ereignissen). In dieser Sprache ist es nie notwendig, „Prioritäten" oder „Verzögerungen" zwischen Übergängen zu benutzen.

Die AEFD-Sprache nimmt den Formalismus des vorangegangenen Abschnitts wieder auf, der sich durch folgende Punkte von einer gewöhnlichen Sprache unterscheidet:

- die Einführung einer Graphenformulierung in Form einer „6 Zeilen" -Textdatei
- die Einführung des Indikatorkonzeptes. Diese werden NACH Ende der Markierungsänderungen aktualisiert
- die Zeit spielt keine Rolle in den Graphen, nur das Ereignis „Verzögerungsende" wird benützt
- die Einführung externer und interner Ereignisse, um eine deterministische Interpretation zu garantieren.

Notons que certaines règles de conception sont à respecter lors de l'écriture du fonctionnel applicatif (logiciel d'application). Celles-ci garantissent une interprétation correcte des graphes, permettent de ne pas augmenter artificiellement le nombre des états système et de lever toute possibilité d'indécision dans l'interprétation des graphes et la gestion des communications. Les caractéristiques de ce langage AEFD sont à préciser.

### 6.2.4.2 Description et fonctionnement d'un graphe

Un graphe est constitué :

- de places ;
- de transitions reliant les places ;
- de conditions de franchissement des transitions ;
- d'actions associées à ces transitions ;
- d'un marquage. Un "jeton" occupe l'état dans lequel se trouve le graphe. Dans la suite, on ne s'intéressera qu'aux graphes dont le marquage est unique.

Ces graphes permettent de décrire les fonctions de signalisation, comme ici la commande d'un sémaphore (Figure 6.6 et Annexe 1).

Einige Konzeptionsregeln sind bei der Formulierung der anwendungsbezogenen Funktionen zu beachten. Diese garantieren eine richtige Interpretation der Graphen, ohne die Anzahl der Systemzustände künstlich zu erhöhen, und beseitigen jede Unentscheidbarkeit bei der Interpretation der Graphen und der Verwaltung der Mitteilungen. Die Eigenschaften dieser AEFD-Sprache werden nun noch präzisiert.

### 6.2.4.2 Beschreibung und Funktionsweise der Graphen

Ein Graph besteht aus:

- Plätzen
- Transitionen, die die Plätze verbinden
- Bedingungen für die Transitionen
- an die Transitionen gebundene Handlungen
- Markierungen. Eine Marke besetzt den Zustand, in dem der Graph sich befindet. Von jetzt an werden nur noch Graphen betrachtet, bei denen die Markierung eindeutig ist.

Diese Graphen erlauben es, die Signalfunktionen zu beschreiben, wie z. B. das Bedienen eines Lichtsignals (Semaphor) (Abbildung 6.6 und Anhang A).

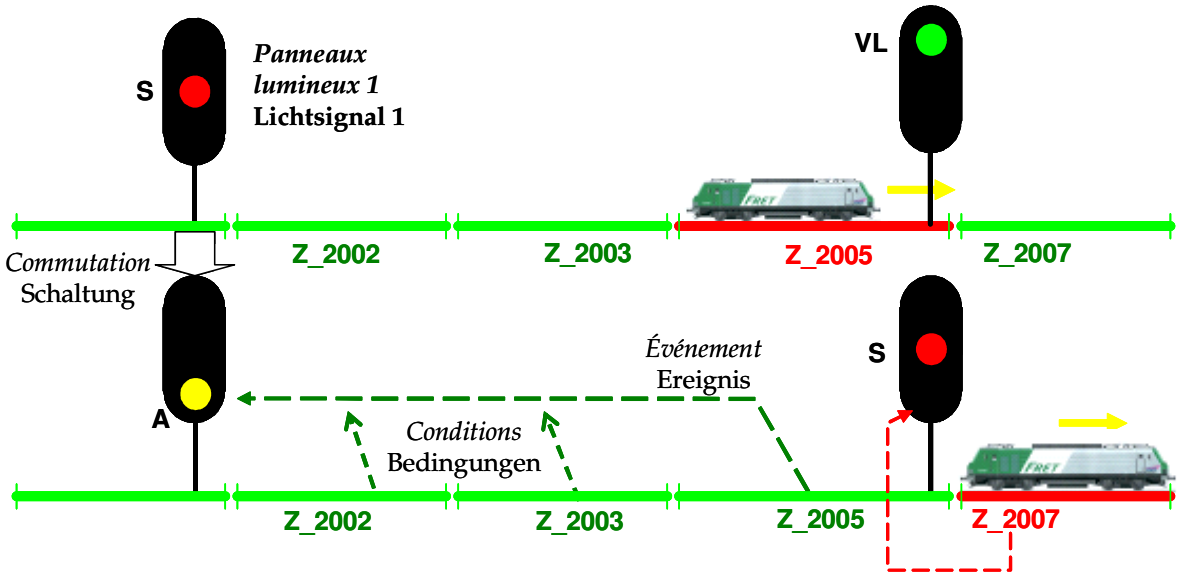



**Figure 6.6 : Changement d'état du panneau lumineux 1 – Commutation S - A**

**Abb. 6.6**: Zustandsänderung des Lichtsignals 1 - Vertauschen von S und A

Cette transition en écriture habituelle [Grude, 1988] [Schnieder, 1992] des réseaux de Petri peut s'écrire sous la forme suivante :

Diese Transition kann in der klassischen Formulierung von Petrinetzen [Grude, 1988] [Schnieder, 1992] wie in Abb. 6.7 geschrieben werden.

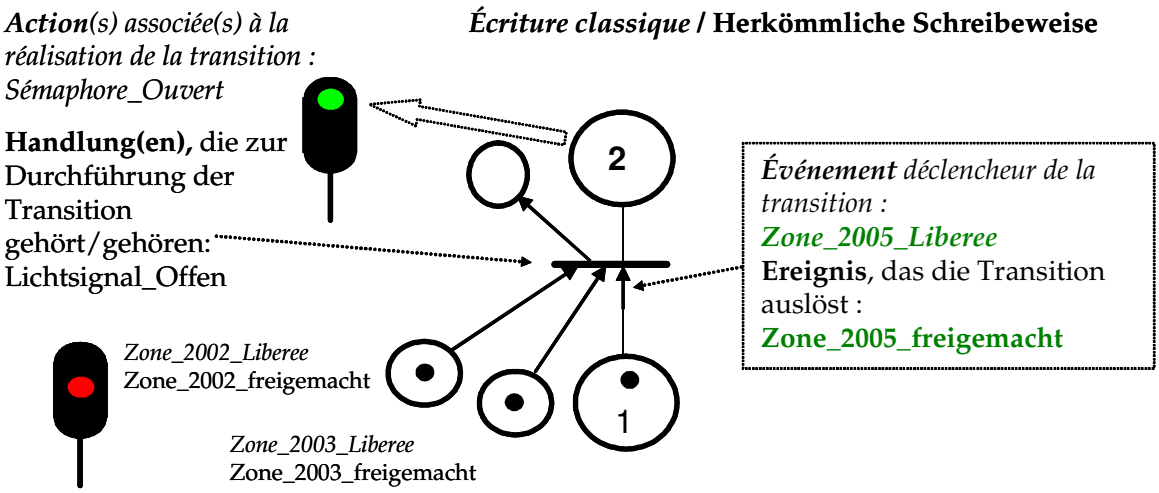
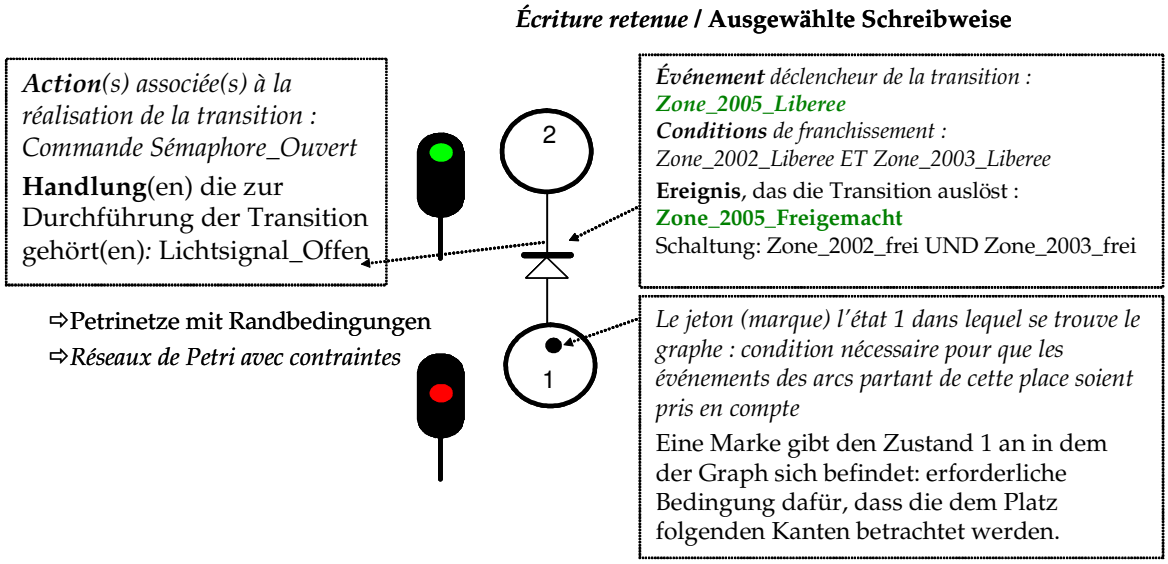



**Figure n°6.7** : Transition S vers A - Écriture classique des réseaux de Petri
**Abbildung n°6.7**: Übergang von S nach A – klassische Schreibweise von Petrinetzen




| *Place origine* **Herkunft platz** | *Place destination* **Zielplatz** | *Événement* **Ereignis** | *Conditions* **Bedingung** | *Action* **Handlung** |
|---|---|---|---|---|
| **1** | **2** | **Zone_2005_Liberee**<br>**Zone_2005_Freigemacht** | Zone_2002_Liberee ET Zone_2003_Liberee<br>Zone_2005_Freigemacht UND Zone_2003_Freigemecht | Semaphore_Ouvert<br>Lichtsignal_Offen |

**Figure 6.8** : Langage AEFD - Notations décrivant les transitions et les marquages
**Abbildung 6.8**: AEFD-Sprache – Schreibweise, die die Transitionen und die Markierungen beschreibt

### 6.2.4.3 Description d'une transition

Un exemple peut être considéré (Figure 6.7):

### 6.2.4.3 Beschreibung einer Transition

Hierfür wird ein Beispiel betrachtet (Abb. 6.7)

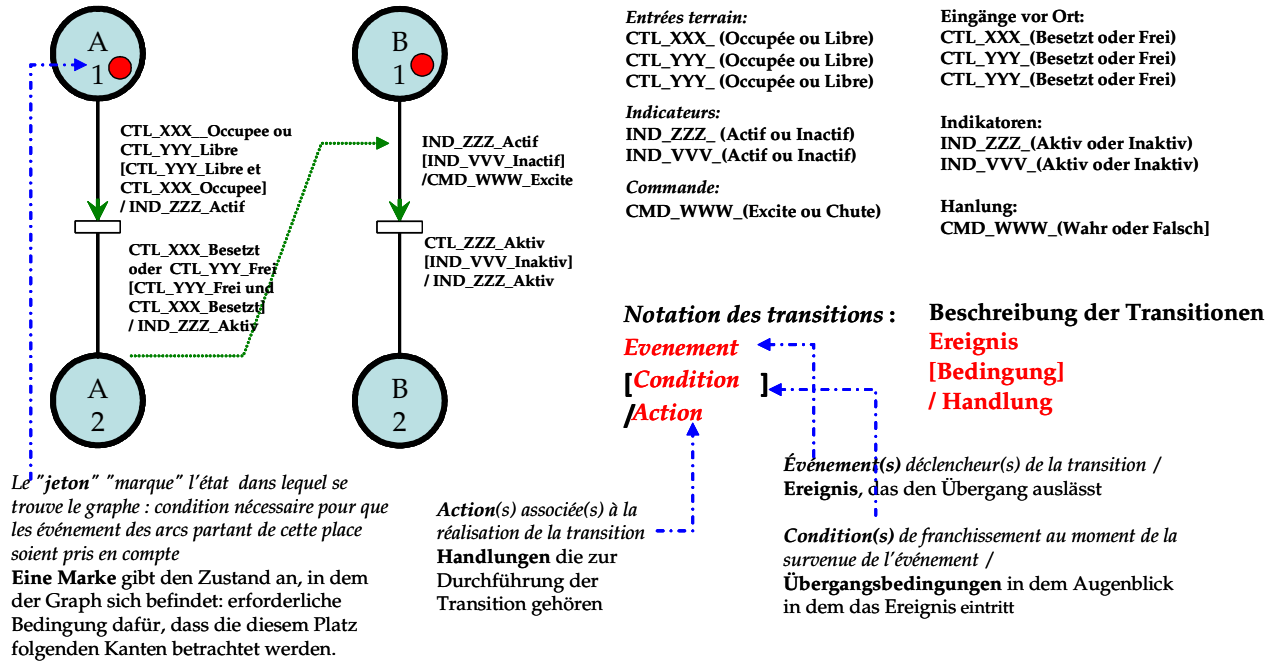



**Figure 6.9** : Traitement d'un événement intervenant sur deux graphes
**Abbildung 6.9**: Behandlung eines Ereignisses, das auf zwei Graphen stattfindet

Les graphes fonctionnent de la manière suivante :

- Un graphe peut se trouver dans un certain nombre d'états, qui sont représentés par des places ;
- Lorsque le graphe se trouve dans un état particulier, la place représentant cet état est marquée (ici le graphe A). A chaque instant, une et une seule place d'un graphe peut être marquée ;
- Les transitions relient entre elles les places du graphe. Elles permettent de transmettre le jeton d'une place à une autre et ainsi de faire évoluer le marquage. Le franchissement d'une transition est associé à l'apparition d'un événement et au respect de conditions. Pour que la transition puisse être franchie, il faut que la place précédant la transition soit marquée et que les conditions associées au franchissement soient vérifiées. Lorsque le marquage franchit une transition, les actions associées à cette transition sont réalisées.

La description du fonctionnement de l'applicatif (logiciel d'application) peut toujours être exprimée sous forme de graphes, indépendamment de la solution technique retenue pour le réaliser.

Die Graphen funktionieren folgendermaßen:

- Ein Graph kann nacheinander mehrere Zustände haben, die durch Plätze dargestellt werden.
- Wenn der Graph sich in einem bestimmten Zustand befindet, wird der Platz, der diesen Zustand darstellt, markiert (im Fall der Abb. 6.9 der Graph A). In jedem Augenblick kann nur ein Zustand eines Graphen markiert sein.
- Die Transitionen verbinden die Zustände der Graphen untereinander. Sie erlauben den Übergang einer Marke von einem Zustand zu einem anderen. So entwickelt sich die Markierung. Die Überschreitung einer Transition ist mit dem Auftreten eines Ereignisses und mit dem Erfüllen der Bedingungen verbunden. Damit die Transition geschaltet werden kann, ist es notwendig, dass der Zustand, der der Transition vorausgeht, markiert ist und dass die mit der Schaltung assoziierten Bedingungen erfüllt sind. Wenn die Markierung einen Übergang macht, werden die mit diesem Übergang assoziierten Handlungen verwirklicht.

Die Beschreibung der Funktionen der Anwendung (Anwendungssoftware) kann immer in Form von Graphen ausgedrückt werden, unabhängig von der technischen Lösung, die gewählt wurde, um die Anwendung zu verwirklichen.

### 6.2.4.4 L'écriture texte (6 lignes)

Afin de permettre leur interprétation en temps réel par une machine cible, les graphes sont regroupés dans un fichier texte unique. Ce fichier décrit ainsi l'automate applicatif complet. Les graphes y sont décrits dans un ordre quelconque, sans effet sur le déterminisme de leur interprétation. Chaque graphe est constitué d'un nombre fini de transitions entre ses places. Chaque transition est décrite par six lignes aux significations suivantes :

1. nom du graphe ;
2. place de départ de la transition ;
3. place d'arrivée de la transition ;
4. ligne texte traduisant une expression booléenne et se terminant par «ÉVENEMENT» ;
5. ligne texte traduisant une expression booléenne et se terminant par «CONDITION»,
6. ligne texte traduisant une liste ordonnée d'actions à réaliser, séparées par des « ; » et se terminant par «ACTION».

### 6.2.4.4 Formulierung in Textform (6 Zeilen)

Um ihre Echtzeitinterpretation durch eine Zielmaschine zu ermöglichen, werden die Graphen in einer einzigen Textdatei zusammengefasst. Diese Datei beschreibt den vollständigen anwendungsspezifischen Automaten. Die Graphen werden dort in einer beliebigen Reihenfolge beschrieben was keinen Einfluss auf die deterministische Interpretation hat. Jeder Graph besteht aus einer endlichen Anzahl von Transitionen zwischen seinen Zuständen. Jede Transition wird durch sechs Zeilen mit der folgenden Bedeutung beschrieben:

1. Name des Graphen
2. Startzustand des Graphen
3. Ankunftzustand des Graphen
4. Textzeile, die einen booleschen Ausdruck übersetzt und die mit dem Wort „EREIGNIS" endet
5. Textzeile, die einen booleschen Ausdruck übersetzt und mit dem Wort „BEDINGUNG" endet
6. Textzeile, die eine geordnete Liste mit zu erfüllenden Handlungen darstellt, getrennt durch ein „;". Die Zeile endet mit dem Wort „HANDLUNG".

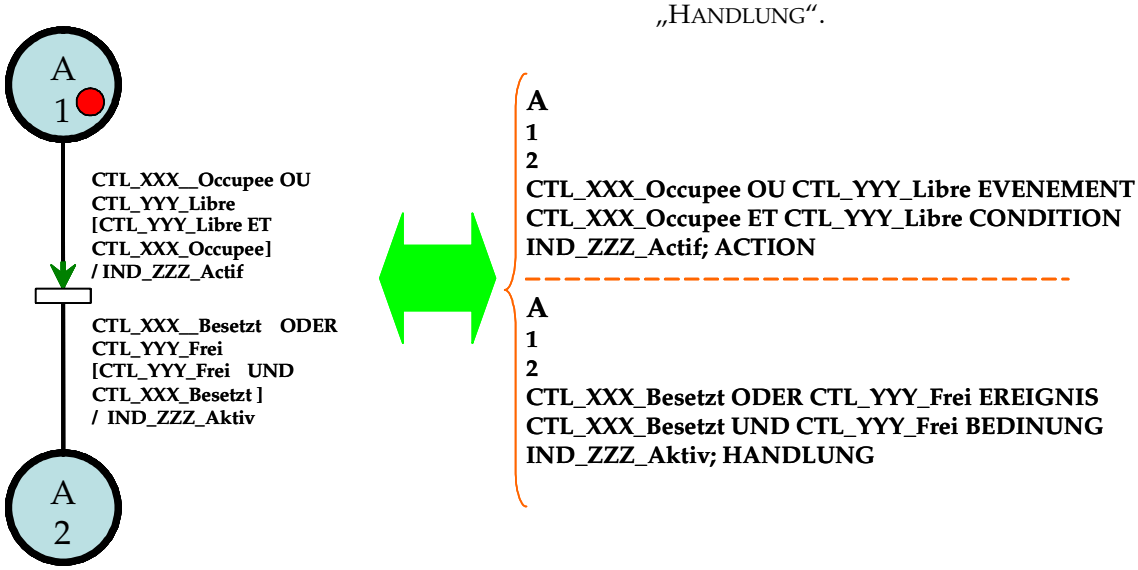



**Figure 6.10** : Exemple d'équivalence entre le formalisme graphique et l'écriture en fichier 6 lignes
**Abb. 6.10**: Gleichwertigkeit Beispiel zwischen dem grafischen Formalismus und der Formulierung als 6-Zeilen Textdatei

Il s'agit sur l'exemple précédent de la description d'une transition du graphe «A» orienté de sa place 1 à sa place 2. La transition est éligible si la place 1 est marquée.
Le franchissement de la transition ne peut être déclenché que, si et seulement si, l'un des deux événements suivant s'est produit : le passage de l'état «Occupée» de l'entrée terrain définie comme CTL_XXX_ ou le passage à l'état Libre de l'entrée terrain CTL_YYY_ (événements externes ici).

Bei dem vorgestellten Beispiel (Abb. 6.10) handelt es sich um die Beschreibung einer Transition von Zustand 1 in den Zustand 2 des gerichteten Graphen A. Die Transition ist schaltfähig, wenn der Zustand 1 markiert ist. Die Transition wird nur dann ausgelöst, wenn eines der beiden folgenden Ereignisse eingetreten ist: der Übergang in den Zustand „besetzt" des externen Eingangs (des Gleisabschnitts), der als CTL_XXX_ definiert ist, oder der Übergang des externen Eingangs (des Gleisabschnitts) CTL_YYY in den freien Zustand (beides sind externe Ereignisse).

La condition de franchissement de la transition ne peut être évaluée favorablement que, si et seulement si, l'équation booléenne est égale à VRAI. C'est-à-dire dans notre cas, l'état Occupée de l'entrée terrain définie comme CTL_XXX_ ET l'état Libre de l'entrée CTL_YYY_ sont Vraies.

Le franchissement de cette transition entraîne la commande ordonnée de l'action suivante, le passage de l'état Inactif à celui Actif de la variable interne définie comme IND_ZZZ_Actif (événement interne). Il est important de noter que le processus d'évaluation est à opérer sur l'ensemble des transitions de l'automate, ce avant de mettre à jour des indicateurs correspondants et, ensuite, de mettre à jour les marquages.

Nous obtenons des interprétations équivalentes sur les plans :
- mathématique : nous pouvons considérer que les transitions internes se réalisent dans un temps nul (transitions internes immédiates),
- physique : les événements externes sont traités de manière ininterruptible selon les modalités définies plus avant.

### 6.2.4.5 Les indicateurs

Une attention particulière doit être portée sur les « indicateurs ». Ceux-ci permettent de distinguer l'évolution des marquages au cours du traitement d'un événement et l'état qu'avait l'automate au moment où est survenu l'événement en cours de traitement. Ils constituent une différence fondamentale par rapport aux réseaux de Petri classiques et ceux disponibles sur le marché. Les indicateurs permettent notamment de se passer de toute «définition de priorités d'évaluation» des transitions, de ne pas être sensible à l'ordre d'évaluation des graphes de l'automate. Ces points sont primordiaux pour rendre l'exécution de l'automate déterministe.

Leur usage doit répondre à certaines règles d'écriture des graphes fonctionnels :
- un indicateur n'est mis à jour que par un unique graphe ;
- un état de l'indicateur est fixé pour un ensemble de places du graphe, l'état complémentaire est imposé pour les autres places du graphe ;
- un indicateur peut être exploité en ÉVENEMENT ou en CONDITION par tous les autres graphes de l'automate.

Remarque : l'existence d'indicateurs avec leurs règles de mis à jour constitue une différence fondamentale avec les réseaux de Petri, y compris les SIPN

Die Bedingung der Schaltung der Transition ist nur dann erfüllt, wenn die boolesche Gleichung WAHR ist. Das heißt in diesem Fall, der Zustand BESETZT des externen Eingangs CTL_XXX_ UND der Zustand FREI des CTL_YYY_-Eingangs sind WAHR.

Die Schaltung der Transition bewirkt die geordnete Steuerung der folgenden Handlung, nämlich den Übergang der als IND_ZZZ_Aktiv definierten internen Variablen vom inaktiven in den aktiven Zustand (internes Ereignis). Es ist wichtig anzumerken, dass der Auswertungsvorgang alle Transitionen des Automaten bearbeitet, bevor die entsprechenden Indikatoren aktualisiert werden. Erst danach werden die Markierungen aktualisiert.

Man erhält so die folgenden Interpretationen:
- mathematisch gesehen: Man kann annehmen, dass die internen Transitionen ohne Zeitverlust stattfinden (unmittelbare interne Übergänge).
- physisch gesehen: Die externen Ereignisse werden ununterbrechbar nach den weiter oben definierten Modalitäten behandelt.

### 6.2.4.5 Die Indikatoren

Die Indikatoren erfordern eine besondere Beachtung. Sie erlauben es, zwischen der Entwicklung der Markierungen während der Behandlung eines Ereignisses und dem Zustand des Automaten zu unterscheiden, den dieser zum Zeitpunkt, als das gerade behandelte Ereignis auftrat, hatte. Sie stellen einen grundlegenden Unterschied zu den herkömmlichen Petrinetzen dar. Die Indikatoren erlauben es, auf Definition von Prioritäten bei der Bearbeitung der Übergänge zu verzichten und nicht von der Bearbeitungsreihenfolge der Graphen des Automaten abhängig zu sein. Diese Punkte sind von vorrangiger Bedeutung, um die Ausführung des Automaten deterministisch zu machen.

Ihr Gebrauch muss bestimmten Formulierungsregeln der funktionellen Graphen entsprechen:
- Ein Indikator wird nur durch einen eindeutigen Graphen aktualisiert.
- Ein Zustand des Indikators wird für eine Gesamtheit von Plätzen des Graphen festgelegt, allen anderen Plätzen wird der entgegengesetzte Zustand zugeordnet.
- Ein Indikator kann für ein EREIGNIS oder eine BEDINGUNG durch alle anderen Graphen des Automaten genutzt werden.

Bemerkung: Die Existenz von Indikatoren und ihren Aktualisierungsregeln stellt den Hauptunterschied zu den normalen Petrinetzen und auch zu den SIPN dar.

Illustrons le fonctionnement d'un graphe théorique utilisant un indicateur (convention de notation inchangée).

Remarques : Si deux transitions partent d'un même état d'un automate, les conditions de franchissement associées à ces deux transitions doivent être exclusives. Au démarrage du système, une place particulière est marquée (dans notre cas la place A1).

In Abb. 6.11 wird das Funktionieren eines theoretischen Graphen mit einem Indikator illustriert (unveränderte Schreibweise).

Bemerkung: Wenn zwei Transitionen vom gleichen Zustand des Automaten ausgehen, müssen deren Übergangsbedingungen ausschließend sein. Beim Systemstart wird ein bestimmter Platz markiert (Platz A1 im Beispiel).

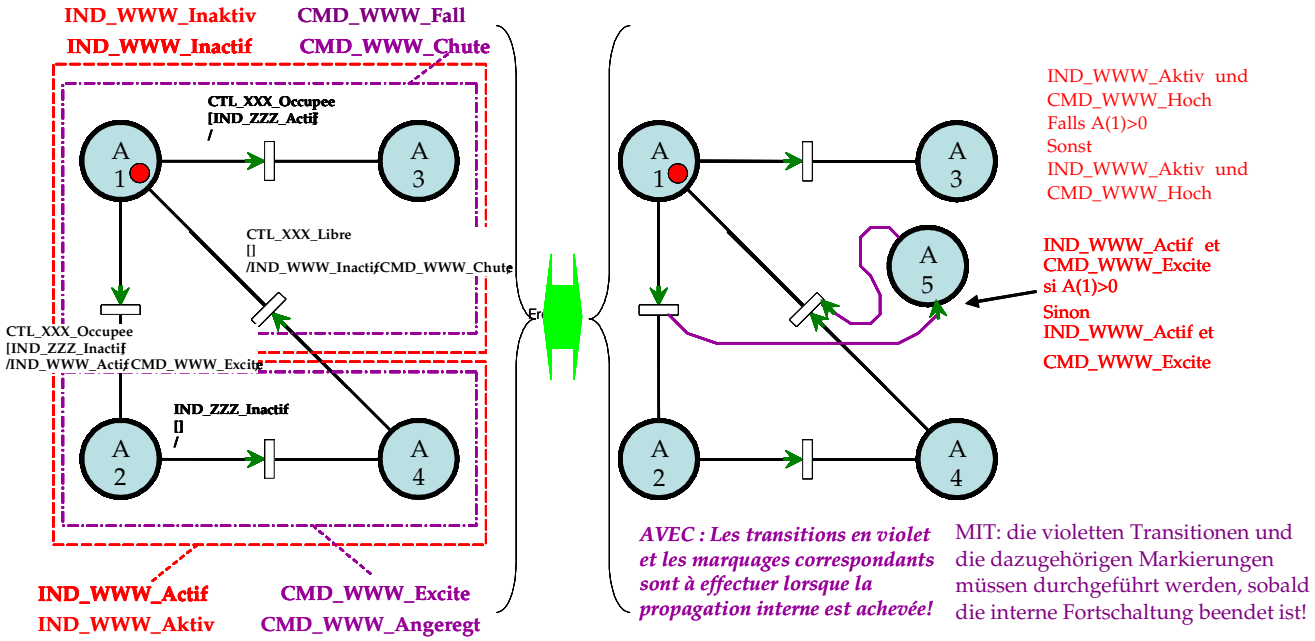



**Figure n°6.11** : Langage AEFD – Gestion des indicateurs et des commandes
**Abbildung n°6.11**: AEFD-Sprache - Verwaltung der Indikatoren und der Aufträge

Toutes les transitions aboutissant à une même place doivent positionner dans le même état l'indicateur qui lui est assigné. Toute commande externe du système est associée à un indicateur et est toujours commandée juste après lui, dans la même ligne action de la même transition. Ainsi sur le graphe AEFD de la figure 6.11 :

- les transitions aboutissant aux places 1 ou 3 portent dans leur champ ACTION le positionnement à 1 de l'indicateur IND_VWW_Inactif et de la commande CMD_WWW_Chute ;
- les transitions aboutissant aux places 2 ou 4 portent dans leur champ ACTION le positionnement à 1 de l'indicateur IND_WWW_Actif et de la commande CMD_WWW_Excite.

Alle Transitionen, die zu ein und demselben Zustand führen, müssen den zugewiesenen Indikator in den gleichen Zustand bringen. Jeder externe Anstoß des Systems ist mit einem Indikator verbunden und wird immer sofort nach dem Indikator in der gleichen Handlungszeile der gleichen Transition aktiviert. Für die Abb. 6.11 bedeutet dies:

- Die Transitionen, die zu den Zuständen 1 oder 3 führen, tragen in ihrem Feld HANDLUNG das Umschalten nach WAHR des IND_WWW_Inaktiv-Indikators und des Befehls CMD_WWW_Nieder
- Die Transitionen, die zu den Zuständen 2 oder 4 führen, tragen in ihrem Feld HANDLUNG das Umschalten in den Zustand 1 des Indikators IND_WWW_Activ und des Befehls CMD_WWW_Hoch.

La mise à jour des indicateurs ne s'opère qu'une fois que l'événement injecté dans l'automate a totalement été propagé, que les marquages des différents graphes ont été réalisés. L'événement externe CTL_XXX_Occupee conduit aux changements d'états suivants :

- transition A1 vers A2 (IND_ZZZ_Actif = 0 à l'initialisation) et positionnement IND_WWW_Actif = 1 ;
- transition A2 vers A4.

L'événement externe CTL_XXX_Libre conduit alors au changement d'état suivant ;

- transition A4 vers A1 et positionnement IND_WWW_Actif = 0.

L'événement externe CTL_XXX_Occupee conduit alors au changement d'état suivant :

- transition A1 vers A3 (état puits) car l'indicateur IND_ZZZ_Actif = 1.

La figure 6.11 montre que l'on peut se rapprocher des réseaux de Petri classiques dans la mesure où l'on peut retarder conditionnellement la mise à jour des places « indicateur » (cas de la place A(5) par exemple).

### 6.2.4.6 Fonctionnement d'un graphe – Exemple avec temporisation

La figure 6.12 illustre le graphe fonctionnel d'une protection ZEP 5028 permettant la gestion d'une protection du personnel. L'événement MSG_31 commande sa mise en action, la succession des événements MSG_32 et MSG_33 dans un court intervalle de temps commande sa levée. L'écoulement d'un délai avant la survenue de l'événement MSG_33 annule l'action antérieure de l'événement MSG_32.

Le graphe correspondant peut ainsi être décrit sous la forme d'un fichier 6 lignes.

Les délais sont gérés de manière extérieure aux graphes eux-mêmes. Le lancement d'une temporisation et l'arrêt d'une temporisation en cours sont des actions. L'échéance d'une temporisation est un événement externe au même titre qu'un changement d'entrée. Cet exemple va permettre de mettre en œuvre les indicateurs et la gestion du temps.

Die Aktualisierung der Indikatoren findet erst dann statt, wenn das in den Automaten eingegebene Ereignis vollständig weitergegeben und die Markierung der verschiedenen Graphen durchgeführt wurde. Das externe Ereignis CTL_XXX_Besetzt, führt zu folgenden Zustandsänderungen:

- Transition A1 nach A2 (IND_ZZZ_Aktiv = 0 bei der Initialisierung) und Stellen von IND_WWW_Aktiv auf 1
- Transition von A2 nach A4.

Das externe Ereignis CTL_XXX_Frei führt dann zu folgender Zustandsänderung:

- Transition A4 nach A1 und Umsetzen von IND_WWW_Aktiv = 0.

Das externe Ereignis CTL_XXX_Belegt führt dann zu folgender Zustandsänderung:

- - Transition A1 nach A3 (absorbierender Zustand) da der Indikator IND_ZZZ_Aktiv auf 1 gesetzt ist.

Abbildung 6.11 zeigt, dass man sich klassischen Petrinetzen annähern kann, sofern man unter bestimmten Bedingungen die Aktualisierung der „Indikatorplätze" verzögern kann (Platz A5 im Beispiel).

### 6.2.4.6 Funktionsweise der Graphen – Beispiel mit Verzögerung

Abbildung 6.12 illustriert den funktionellen Graph eines ZEP 5028 zur Verwaltung eines Personenschutzes. Das Ereignis MSG_31 leitet den Schutz ein, das Eintreten von MSG_32 und MSG_33 innerhalb eines kurzen Zeitintervalls hebt ihn auf. Das Ablaufen einer Frist vor dem Ereignis MSG_33 annulliert die frühere Aktion des Ereignisses MSG_32.

Der entsprechende Graph kann so in der Form einer 6-Zeilendatei formuliert werden.

Die Fristen werden außerhalb der Graphen verwaltet. Die Initiierung und das Aufheben einer laufenden Verzögerung sind Handlungen. Das Fälligkeitsdatum einer Verzögerung ist ein externes Ereignis ebenso wie eine Eingangsänderung. Dieses Beispiel erlaubt es, die Indikatoren und die Zeitverwaltung einzusetzen.

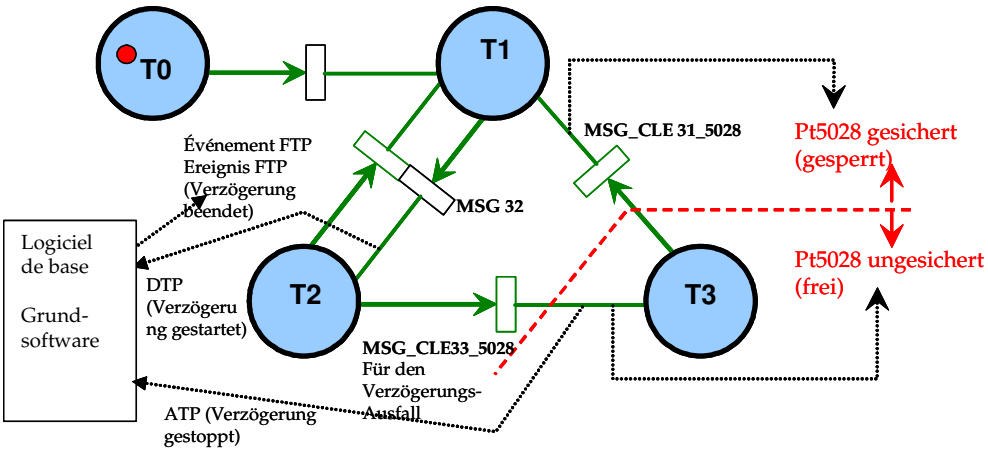



**Figure 6.12** : Graphe PZP5028 pour la protection du personnel en voie /
**Abbildung 6.12:** PZP5028 Graph für den Personenschutz in einer Zone

*Nota : Une transition est représentée avec une seule flèche ce qui n'est pas représentation habituelle*

**Exemple d'écriture du graphe assurant de la gestion de la protection PZP5028 :**

Transition T0 T1
Événement ACT_Init_ZEP_5028
Condition IND_Pt5028_Assuree
Action

Transition T1 T2
Événement MSG_CLE32_5028
Condition IND_RIt2002_2001_Inactif ET IND_CIt2004_2001_Inactif
Action DTP_CLE32_5028 : 300

Transition T2 T1
Événement FTP_CLE32_5028
Condition
Action

Transition T2 T3
Événement MSG_CLE33_5028
Condition IND_RIt2002_2001_Inactif ET IND_CIt2004_2001_Inactif
Action ATP_CLE32_5028; IND_Pt5028_non_Assuree;

Transition T3 11
Événement MSG_CLE31_5028
Condition IND_RIt2002_2001_Inactif ET IND_CIt2004_2001_Inactif
Action IND_Pt5028_Assuree

## 6.2.4.7 Graphes traduisant les temporisations fonctionnelles

D'un point de vue fonctionnel, quatre cas sont à considérer pour chaque temporisation. Ils correspondent chacun à un état stable :

*Nota: Eine Schaltung ist mit einem einzigen Pfeil illustriert. Dies ist nicht die gewöhnliche Darstellung*

**Beispiel der Formulierung eines Graphen der den Schutz auf der Gleiszone PZP 5028 sichert:**

Transition T0 T1
Ereignis ACT_Init_ZEP_5028
Bedingung IND_Pt5028_Gesichert
Handlung

Transition T1 T2
Ereignis MSG_CLE32_5028
Bedingung IND_RIt2002_2001_Inaktiv UND IND_CIt2004_2001_Inaktiv
Handlung DTP_CLE32_5028 : 300

Transition T2 T1
Ereignis FTP_CLE32_5028
Bedingung
Handlung

Transition T2 T3
Ereignis MSG_CLE33_5028
Bedingung IND_RIt2002_2001_Inaktiv UND IND_CIt2004_2001_Inaktiv
Handlung ATP_CLE32_5028; IND_Pt5028_nicht_gesichert;

**Transition** T3 T1
Ereignis MSG_CLE31_5028
Bedingung IND_RIt2002_2001_Inaktiv UND IND_CIt2004_2001_Inaktiv
Handlung IND_Pt5028_Gesichert;

## 6.2.4.7 Graphen, die die funktionelle Verzögerung darstellen

Vom funktionellen Standpunkt aus sind bei jeder Verzögerung vier Fälle zu betrachten. Jeder entspricht einem stabilen Zustand:

1. La temporisation n'est pas armée. Aucun événement ne se produit ;
2. La temporisation est armée par un automate des graphes fonctionnels. Une action de type «DTP» est réalisée. La temporisation est en cours. Quelque soit la valeur courante de la temporisation, il s'agit d'un état dans lequel la temporisation a été armée mais n'est pas encore arrivée à échéance. Par exemple pour une temporisation de 3 minutes, qu'il se soit écoulé de 1 ou 2 minutes, seul l'état courant de la temporisation (temporisation en cours) importe, la valeur précise de la temporisation n'a pas de signification fonctionnelle ;
3. La temporisation en cours est échue. L'événement de type «FTP» est généré par le logiciel de base et est injecté dans les graphes fonctionnels ;
4. La temporisation en cours est arrêtée par un automate des graphes fonctionnels (quelque soit la valeur de la temporisation). L'action de type «ATP» est réalisée. Le logiciel de base supprime la temporisation associée.

Exemple sur l'automate PZP 5028

| | |
|---|---|
| *Transition* | T0 T1 |
| *Événement* | ACT_Init_ZEP_5028 |
| *Condition* | IND_Pt5028_Assuree |
| *Action* | |

Remarque : initialisation du graphe à la mise en service

| | |
|---|---|
| *Transition* | T1 T2 |
| *Événement* | MSG_CLE32_5028 |
| *Condition* | IND_RIt2002_2001_Inactif ET<br>IND_RIt2004_2001_Inactif ET<br>IND_CIt2002_2001_Inactif ET<br>IND_CIt2004_2001_Inactif |
| *Action* | **DTP_CLE32_5028 : 300** |

Remarque : à partir de l'état initial, il n'est pas possible d'atteindre la place T2 sans passer par la place T1 (Figure 6.13) dans laquelle aucune temporisation n'est en cours. Dans la place T2, la temporisation est en cours suite à l'action d'armement de la temporisation DTP_CLE32_5028 : 300 (valeur 300 * 0,1 s).

| | |
|---|---|
| *Transition* | T2 T1 |
| *Événement* | FTP_CLE32_5028 |
| *Condition* | |
| *Action* | |

1. Die Verzögerung ist nicht aktiv, kein Ereignis tritt ein
2. Die Verzögerung wird von einem Automaten funktioneller Graphen aktiviert. Eine Handlung des Typs „DTP" wird durchgeführt. Die Zeitverzögerung ist aktiv. Unabhängig vom aktuellen Wert der Verzögerung handelt es sich um einen Zustand, in dem die Verzögerung aktiv, aber noch abgelaufen ist. Zum Beispiel bei einer Verzögerung von drei Minuten bei der eine oder zwei Minuten abgelaufen sind, ist nur der aktuelle Zustand der Verzögerung (laufende Zeitverzögerung) wichtig, der genaue Wert der ganzen Verzögerung hat keine funktionelle Bedeutung
3. Die aktuelle Zeitverzögerung ist fällig geworden. Das Ereignis des Typs „FTP" wird durch die Grundsoftware hervorgerufen und in die funktionellen Graphen eingegeben.
4. Die laufende Zeitverzögerung ist durch einen Automaten des funktionellen Graphen angehalten worden (unabhängig vom Wert der Zeitverzögerung). Die Handlung des Typs „ATP" wird durchgeführt. Die Grundsoftware löscht die assoziierte Zeitverzögerung.

Beispiel auf dem Automaten PZP 5028

| | |
|---|---|
| *Transition* | T0 T1 |
| *Ereignis* | ACT_Init_ZEP_5028 |
| *Bedingung* | IND_Pt5028_Gesichert |
| *Handlung* | |

Bemerkung: Initialisierung, wenn das System in Betrieb genommen wird.

| | |
|---|---|
| *Transition* | T1 T2 |
| *Ereignis* | MSG_CLE32_5028 |
| *Bedingung* | IND_RIt2002_2001_Inaktiv UND<br>IND_RIt2004_2001_Inaktiv UND<br>IND_CIt2002_2001_Inaktiv UND<br>IND_CIt2004_2001_Inaktiv |
| *Handlung* | **DTP_CLE32_5028 : 300** |

Bemerkung: vom Anfangszustand aus ist es nicht möglich, den Platz T2 zu erreichen, ohne über den Platz T1 (Abb. 6.13) zu gehen, auf dem keine Verzögerung läuft. Auf Platz T2 läuft eine Verzögerung aufgrund der Aktivierung von DTP_CLE32_5028 : 300 (Wert 300 · 0,1 s).

| | |
|---|---|
| *Transition* | T2 T1 |
| *Ereignis* | FTP_CLE32_5028 |
| *Bedingung* | |
| *Handlung* | |

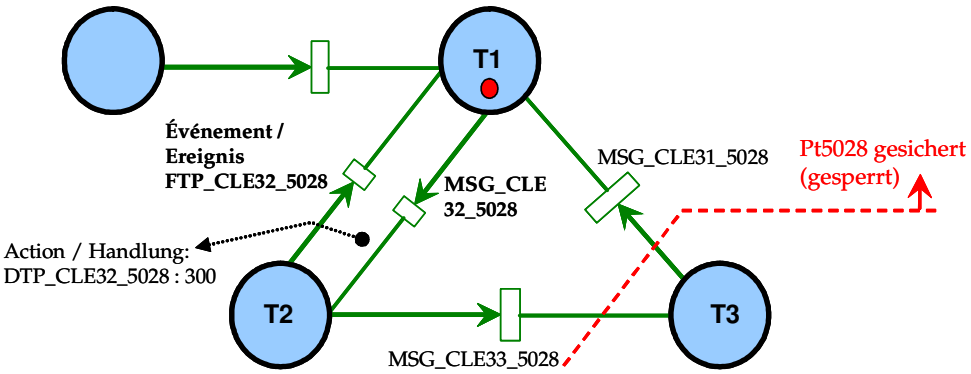



**Figure 6.13** : État courant T1 – Portion de voie protégée

**Abbildung 6.13**: Laufender Zustand T1 – Gleiszone gesperrt

| | |
|---|---|
| *Transition* | *T2 T3* |
| *Événement* | MSG_CLE33_5028 |
| *Condition* | IND_RIt2002_2001_Inactif ET |
| | IND_RIt2004_2001_Inactif ET |
| | IND_CIt2002_2001_Inactif ET |
| | IND_CIt2004_2001_Inactif |
| *Action* | ATP_CLE32_5028; IND_Pt5028_non_Assuree |

Remarque : Dans la place T2 la temporisation est en cours (Figure 6.14). Dans la place T3 la temporisation est arrêtée par l'action ATP_CLE32_5028. La temporisation n'est pas active en place T3.

| | |
|---|---|
| *Transition* | *T3 T1* |
| *Événement* | MSG_CLE31_5028 |
| *Condition* | IND_RIt2002_2001_Inactif ET |
| | IND_RIt2004_2001_Inactif ET |
| | IND_CIt2002_2001_Inactif ET |
| | IND_CIt2004_2001_Inactif |
| *Action* | IND_Pt5028_Assuree |

Remarque : La temporisation n'est pas active en place T3 et en place T1.

| | |
|---|---|
| *Transition* | T2 T3 |
| *Ereignis* | MSG_CLE33_5028 |
| *Bedingung* | IND_RIt2002_2001_Inaktiv UND |
| | IND_RIt2004_2001_Inaktiv UND |
| | IND_CIt2002_2001_Inaktiv UND |
| | IND_CIt2004_2001_Inaktiv |
| *Handlung* | ATP_CLE32_5028; IND_Pt5028_Ungesichert |

Bemerkung: Auf Platz T2 ist eine Verzögerung am Laufen (Abb. 6.14). Auf Platz T3 wird die Verzögerung durch die Handlung ATP_CLE32_5028 angehalten. Die Verzögerung ist auf Platz T3 inaktiv.

| | |
|---|---|
| *Transition* | T3 T1 |
| Ereignis | MSG_CLE31_5028 |
| Bedingung | IND_RIt2002_2001_Inaktiv UND |
| | IND_RIt2004_2001_Inaktiv UND |
| | IND_CIt2002_2001_Inaktiv UND |
| | IND_CIt2004_2001_Inaktiv |
| Handlung | IND_Pt5028_Gesichert |

**Bemerkung: Die Verzögerung ist auf den Plätzen T3 und T1 inaktiv.**

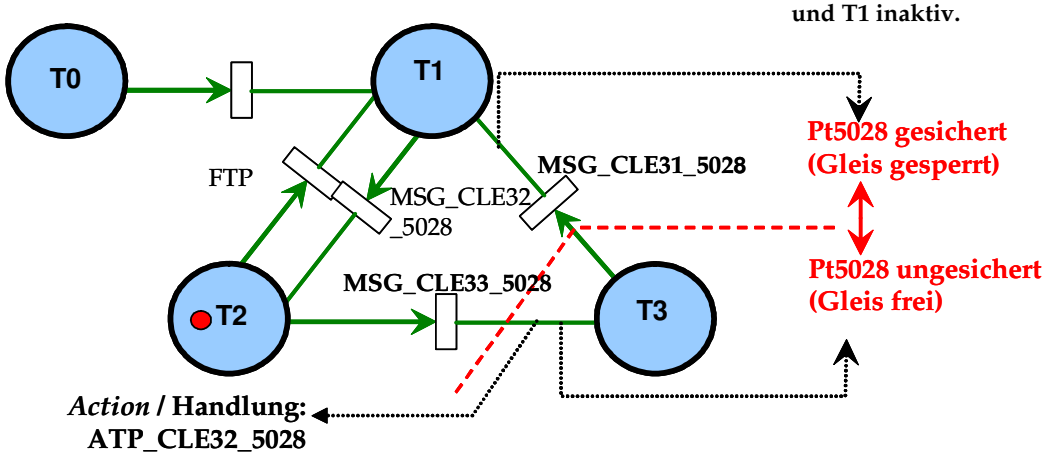



**Figure 6.14** : État courant T2 – Temporisation en cours (délais maximal entre les MSG_CLE32 et MSG_CLE33)

**Abbildung 6.14**: Laufender Zustand T2 –Eine Verzögerung ist am Laufen (maximale Dauer zwischen MSG_CLE32 und MSG_CLE33)

### 6.2.4.8 Conséquences sur l'exploration des états système

Pour la génération des états accessibles, aucun traitement particulier n'est nécessaire pour la gestion des temporisations :

- Les actions de type «ATP» et «DTP» sont à traiter comme les actions de type «CMD». Tout se passe comme s'il s'agissait de la commande d'un minuteur sauf que c'est le logiciel de base qui s'en charge au lieu d'un équipement externe ;
- La gestion du temps (horloge physique) est externe aux graphes ;
- Les événements de type «FTP» sont à traiter comme des événements externes de type «CTL» ;
- Les graphes retenus ne sont pas des graphes d'états temporisés. Ce mode de gestion du temps permet d'obtenir un automate à états finis (Figure 6.15). Ceci sera primordial pour la méthode de preuve retenue.

### 6.2.4.8 Auswirkungen auf die Auswertung der Systemzustände

Bei der Erzeugung der zugänglichen Zustände ist für die Verwaltung der Verzögerungen keine besondere Bearbeitung notwendig.

- Die Handlungen des Typs „ATP“ und „DTP“, werden wie die Handlungen des Typs „CMD“ behandelt. Alles geschieht, als ob es sich um die Bedienung einer Stoppuhr handelt, außer dass statt eines externen Gerätes dafür die Grundsoftware zuständig ist.
- Die Zeitverwaltung (physikalische Uhr) geschieht außerhalb der Graphen.
- Die Ereignisse des Typs „FTP“ werden als externe Ereignisse des Typs „CTL“ behandelt.
- Die ausgewählten Graphen sind keine Graphen mit verzögerten Zuständen. Die hier vorgestellte Art der Zeitverwaltung erlaubt es, endliche Automaten zu erhalten (Abb. 6.15). Dies wird für die ausgewählte Beweismethode von vorrangiger Bedeutung sein.

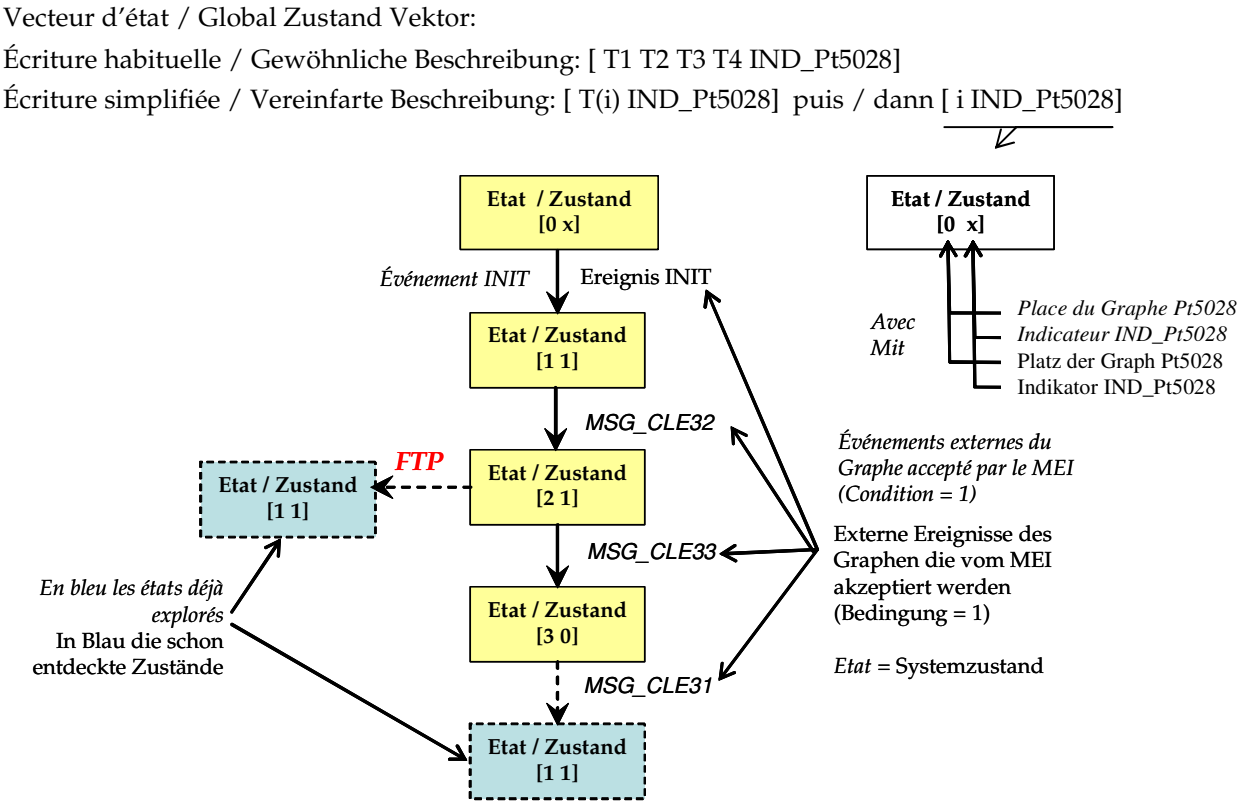



**Figure 6.15** : Arbre des états accessibles du graphe PZP 5028
**Abbildung 6.15**: Erreichbarkeitsbaum für den Graphen PZP 5028

### 6.2.4.9 Illustration de l'Interprétation déterministe

Le déterminisme de l'interprétation d'un réseau d'automates repose sur l'écriture des automates et le mode d'interprétation du réseau d'automates. L'interpréteur de réseau d'automates, également appelé moteur de résolution des graphes, est construit autour de 4 piles de type First In First Out (FIFO). Le principe général est que les éléments d'une pile donnée ne peuvent commencer à être dépilés que si l'ensemble des piles de rangs inférieurs est vide. La pile :

1. contient les événements et actions à usage interne de l'interpréteur ;
2. contient la liste des marquages d'automates devant être mis à jour ;
3. contient la liste des indicateurs à modifier ;
4. contient les événements externes attente d'injection dans le moteur.

Il remarquable de noter que ce mode d'interprétation permet de ne pas être sensible à l'ordre dans lequel les différents automates sont interprétés. Ainsi aucune contrainte n'est posée pour l'écriture des automates[69].

Les avantages de ce type d'interprétation sont :

- l'ajout de graphes au fonctionnel ne nécessite pas la modification des graphes déjà existants. Ce pourrait être le cas si l'ordre de lecture des graphes dépendait de priorités établies lors de l'écriture des graphes ;
- les événements externes sont traités un par un puisqu'il n'est pas possible d'injecter un nouvel événement externe tant que les FIFO 1, 2 et 3 ne sont pas vides et tant que le système n'a pas atteint un état stable. C'est ce point qui rend le fonctionnel d'un PIPC *formellement* prouvable facilement ;
- le fonctionnement obtenu est celui d'un automate à nombre fini d'états et ce quel que soit le nombre de graphes de l'automate globale, de la hiérarchie entre ces graphes… C'est ce qui rend notre démarche originale réalisable.

L'exemple suivant (figure 6.16) présente sur un cas simple le mode d'interprétation des automates.

### 6.2.4.9 Deterministische Interpretation

Der Determinismus der Interpretation eines Automatennetzes beruht auf der Formulierung der Automaten und auf der Methode der Interpretation des Automatennetzes. Der Interpreter des Automatennetzes, der auch Lösungsmotor (oder -maschine) genannt wird, beruht auf vier „First in First out" (FIFO) Stapeln. Der allgemeine Grundsatz dieses Typs von Stapel ist, dass ein Element des Speichers nur dann ausgelesen werden kann, wenn alle Speicher niedrigeren Rangs leer sind. Der Speicher enthält:

1. die Ereignisse und Handlungen für den internen Gebrauch durch den Interpreter.
2. die Liste der Automatenmarkierungen die aktualisiert werden müssen.
3. die Liste der zu ändernden Indikatoren.
4. die externen Ereignisse, die darauf warten, in die Maschine eingespeist zu werden.

Es ist für diese Interpretationsmethode kennzeichnend, dass die Reihenfolge, in der die verschiedenen Automaten interpretiert werden, belanglos ist. So gibt es beim Schreiben der Automaten keine Randbedingungen[70].

Die Vorteile dieser Art von Interpretation sind:

- Das Hinzufügen eines Graphen zu den Funktionen ändert die bestehenden Graphen nicht. Dies kann der Fall sein, wenn die Reihenfolge des Lesens der Graphen von den Prioritäten abhängt, die bei der Formulierung der Graphen festgelegt wurden.
- Die externen Ereignisse werden eines nach dem anderen behandelt, da es nicht möglich ist, ein neues externes Ereignis einzulesen, solange die Speicher FIFO 1, 2 und 3 nicht leer sind und solange das System keinen stabilen Zustand erreicht hat. Dieser Gesichtspunkt macht die Funktionen eines PIPC formal leicht beweisbar.
- Die erhaltene Funktionsweise ist die eines endlichen Automaten und dies unabhängig von der Anzahl der Graphen des kompletten Automaten und von der Hierarchie zwischen diesen Graphen. Dies macht diese originelle Vorgehensweise anwendbar.

Das Beispiel aus Abb. 6.16 stellt für einen einfachen Fall die Automateninterpretationsmethode vor.

[69] Nous nous retrouvons ainsi à nouveau dans les conditions des tables d'enclenchement des poates d'aiguillages mécaniques

[70] So befindet man sich wieder in der Situation der Mechanischen-Stellwerke

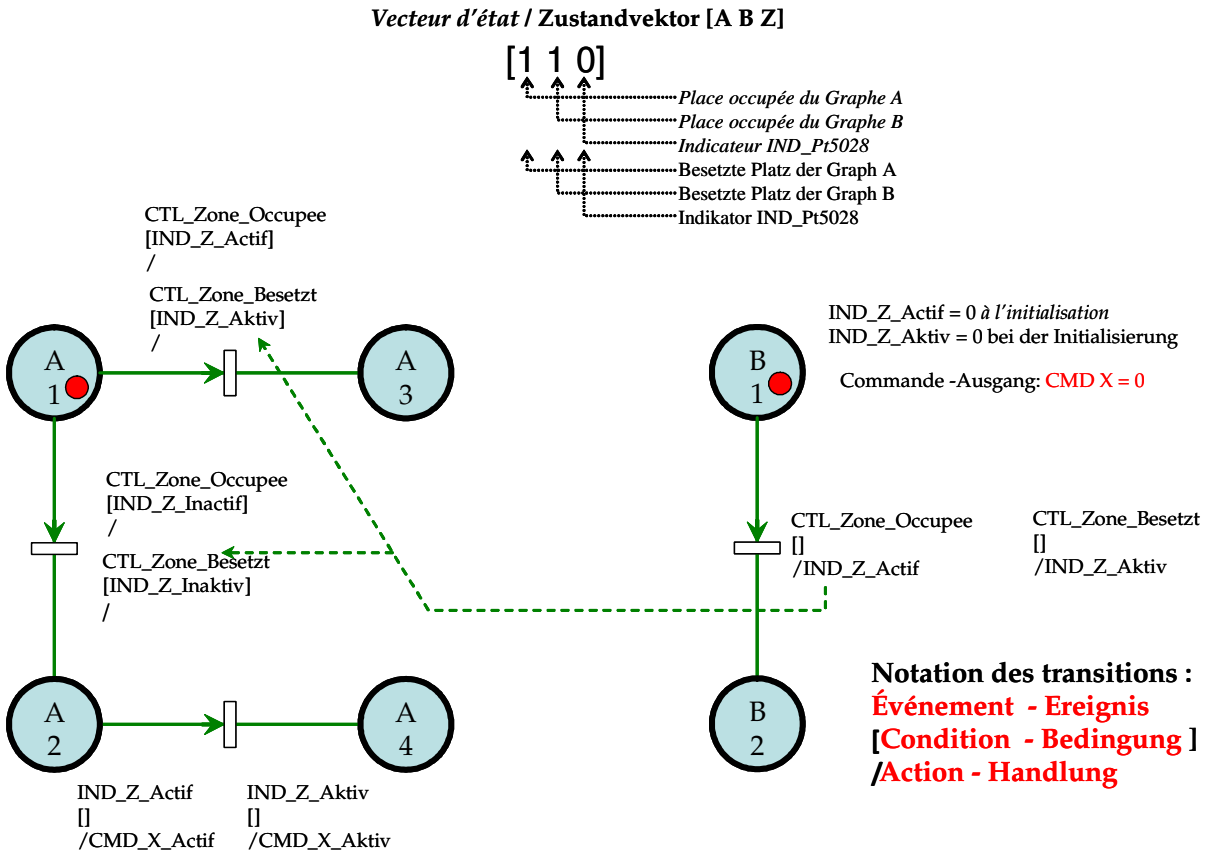



**Figure 6.16** : Exemple d'automate à deux graphes A et B
**Abbildung 6.16**: Beispiel eines Automaten mit zwei Graphen A und B

Les deux graphes A et B sont tous les deux initialisés en place 1. Dans leur état initial, ne sont sensibles qu'à l'événement CTL_Zone_Occupee. L'état des FIFO de l'interpréteur est détaillé ci-dessous au cours du traitement de l'événement externe CTL_Zone_Occupee.

Les figures suivantes (6.17 et 6.18) illustrent le fonctionnement des piles FIFO, notamment quant au traitement différentié entre événements internes et événements externes.

Die zwei Graphen A und B werden beide im Zustand 1 initialisiert. In ihrem Anfangszustand sind sie nur für das CTL_Zone_Besetzt Ereignis empfänglich. Der Zustand des FIFO-Stapels des Interpreters während der Bearbeitung des externen Ereignisses CTL_Zone_Besetzt wird weiter unten im Einzelnen beschrieben. Die Abbildungen 6.17 und 6.18 illustrieren das Funktionieren des FIFO-Stapels, insbesondere bei der unterschiedlichen Bearbeitung der internen und externen Ereignisse.

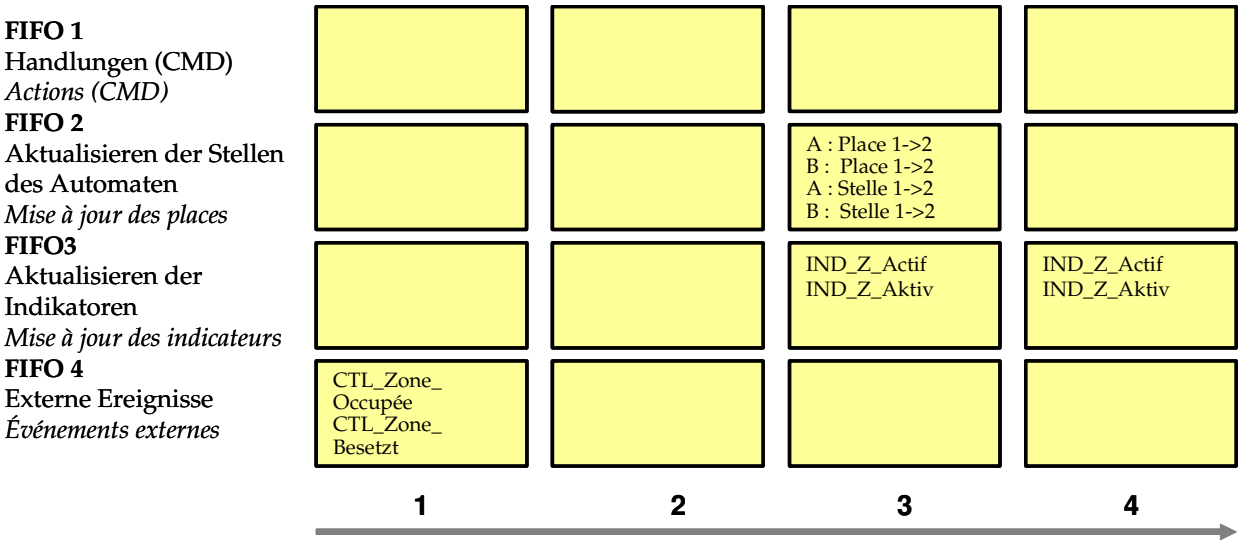



**Figure 6.17** : Fonctionnement de l'interpréteur à pile sur l'exemple de la figure 6.16
**Abbildung 6.17**: Funktionsweise des Stapelinterpreters dargestellt anhand des Beispiels aus Abb. 6.16

Temps 1 : (Figure 6.17) La modification de l'état d'une entrée terrain est détectée (ici une zone devient occupée). L'événement correspondant est injecté dans l'interpréteur.

Temps 2 : L'événement externe est dépilé et injecté dans le moteur d'interprétation.

Temps 3 : Les nouveaux marquages résultant des franchissements de transition que l'événement externe a rendue possibles sont empilés dans la FIFO 2. Le champ action de la transition B :1→2 entraîne la modification de la valeur de l'indicateur Z, cette action est empilée dans FIFO3.

Temps 4 : le nouveau marquage du graphe A puis du graphe B est mis à jour

Temps 5 : (Figure 6.18) l'indicateur IND_Z prend sa nouvelle valeur. Ceci entraîne que la transition A :2 → 4 de l'automate A devient franchissable. Cette transition est associée un champ action commandant une sortie terrain. Cette action est empilée en FIFO 1.

Temps 6 : la sortie est mise à jour.

Temps 7 : le nouveau marquage du graphe A devient effectif.

Ainsi dans le cas de l'exemple précédent nous obtenons la synthèse des états systèmes suivants :

Zeitpunkt 1: (Abb. 6.17) Die Änderung des Zustands eines externen Eingangs wird festgestellt (im vorliegenden Fall wird ein Gleisstromkreis besetzt). Das entsprechende Ereignis wird in den Interpreter eingegeben.
Zeitpunkt 2: Das externe Ereignis wird aus dem Speicher ausgelesen und in den Interpretermotor eingelesen.
Zeitpunkt 3: Die neuen Markierungen, die aus dem Schalten der Transition entstanden sind, welches das externe Ereignis möglich gemacht hat, werden im Speicher FIFO 2 gestapelt. Das Aktionsfeld des Übergangs B:1→2 bewirkt die Änderung des Wertes des Indikators Z; diese Aktion wird im Speicher FIFO 3 gestapelt.
Zeitpunkt 4: Die neue Markierung des Graphen A und B ist gespeichert.

Zeitpunkt 5: (Abb. 6.18) Der IND_Z-Indikator nimmt seinen neuen Wert an. Dies bewirkt, dass die Transition A:2 → 4 des Graphen A schaltbar wird. Diese Transition ist mit einem Aktionsfeld verbunden, das einen Ausgang vor Ort steuert. Diese Aktion wird im FIFO Speicher 1 gestapelt.

Zeitpunkt 6: Die Ausgabe wird aktualisiert.

Zeitpunkt 7: Die neue Markierung des Graphen A wird wirksam.

Im Falle des vorangegangenen Beispiels (siehe Abb. 6.16) erhält man zusammenfassend die Systemzustände, die in Abb. 6.19 dargestellt sind.

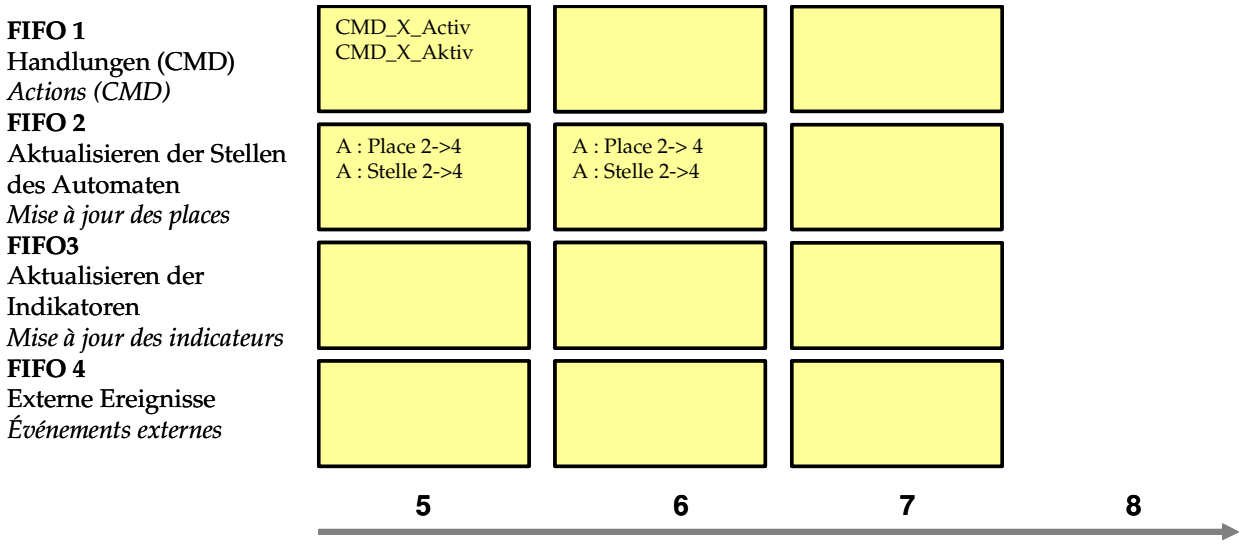



**Figure 6.18** : Fonctionnement de l'interpréteur à pile sur l'exemple de la figure 6.16 (suite)
**Abbildung 6.18**: Funktionsweise des Stapelinterpreters dargestellt anhand des Beispiels aus Abb. 6.16 (Fortsetzung)

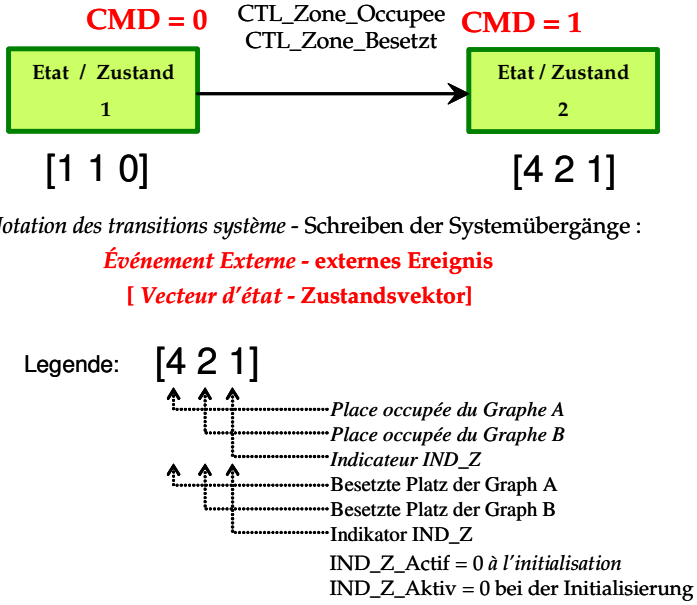



**Figure 6.19** : États systèmes avant et après une transition effectuée par le moteur du PIPC
**Abb. 6.19**: Systemzutände vor und nach einer Transition, die durch die Grundsoftware des PIPC durchgeführt wurde

Les exemples suivants prolongent le cas précédent afin de mieux comprendre le mode d'interprétation des automates et l'impact des règles de conception des graphes afin de maîtriser l'explosion combinatoire.

Die folgenden Beispiele setzen den vorhergehenden Fall fort, um die Methode der Automateninterpretation und die Auswirkung der Konzeptionsregeln der Graphen die die kombinatorische Explosion verhindern, besser darzustellen.

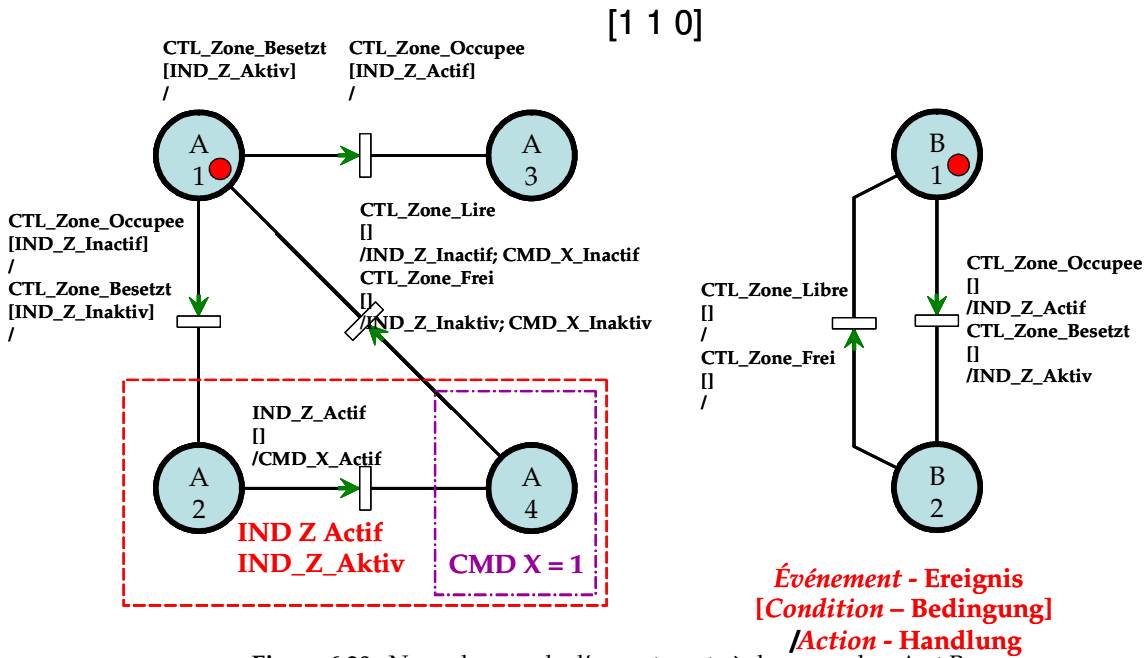



**Figure 6.20** : Nouvel exemple d'un automate à deux graphes A et B
**Abbildung 6.20**: Neues Beispiel eines Automaten mit zwei Graphen A und B

Ainsi dans le cas de l'exemple précédent, l'indicateur (IND) n'est pas strictement associé à une ou plusieurs places d'un graphe, nous obtenons la synthèse des états systèmes illustrés par les figures 6.21 et 22 (Annexe C1).

Ainsi dans le cas de l'exemple de la figure 6.20 précédent nous obtenons les différentes étapes de fonctionnement :

Im Falle des Beispiels aus Abb. 6.20 ist der Indikator (IND) nicht strikt mit einem oder mehreren Zuständen des Graphen verbunden. Man erhält zusammenfassend die Systemzustände die Abb. 6.21 und 6.22 (siehe Anhang C1).

Im Falle des vorangegangenen Beispiels (siehe Abb. 6.20) erhält man zusammenfassend die verschiedenen Situationen:

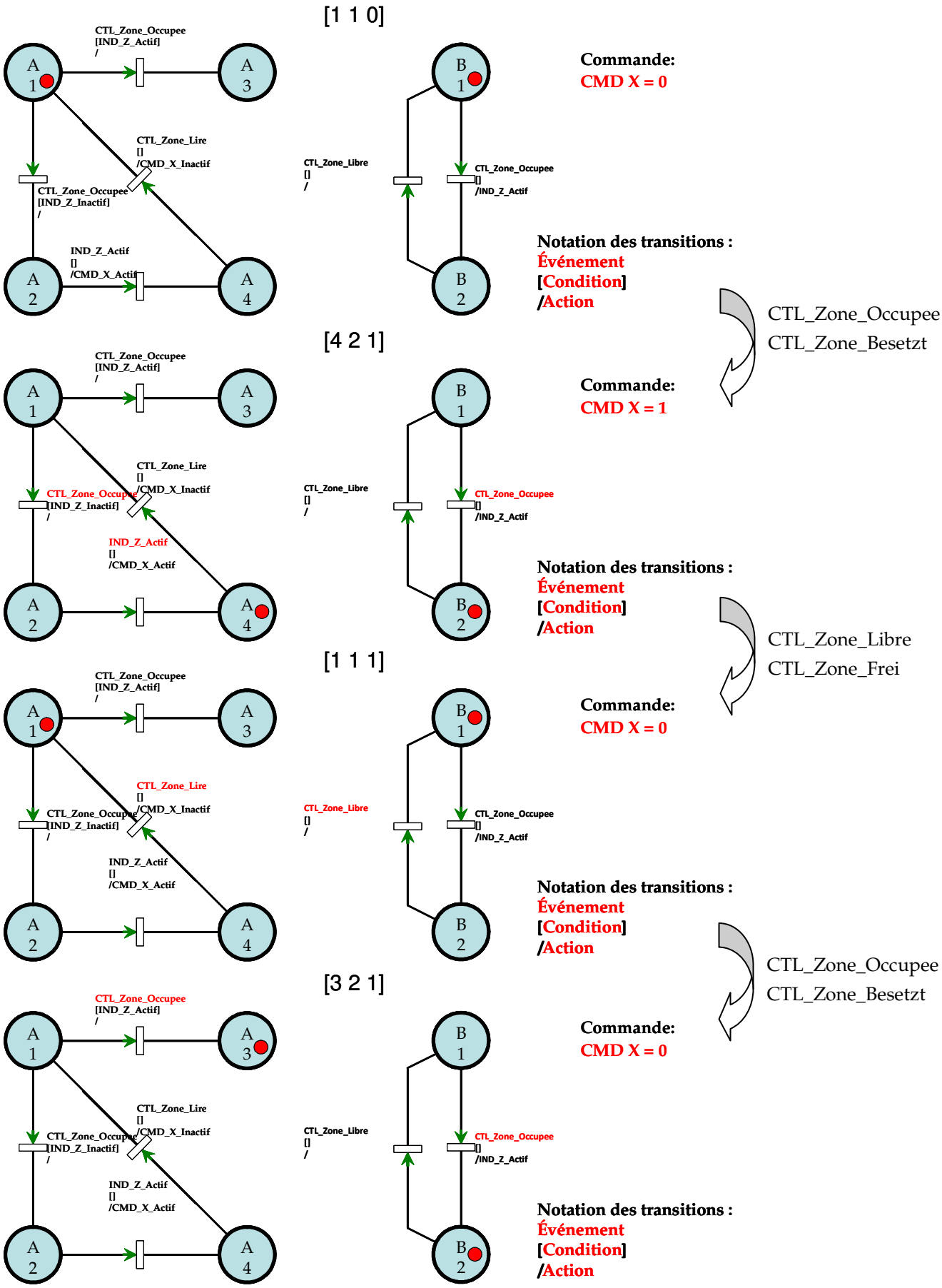



**Figure 6.21** : Fonctionnement de l'exemple de la figure 6.16
**Abbildung 6.21**: Funktionieren des Beispiels in Abb.6.16

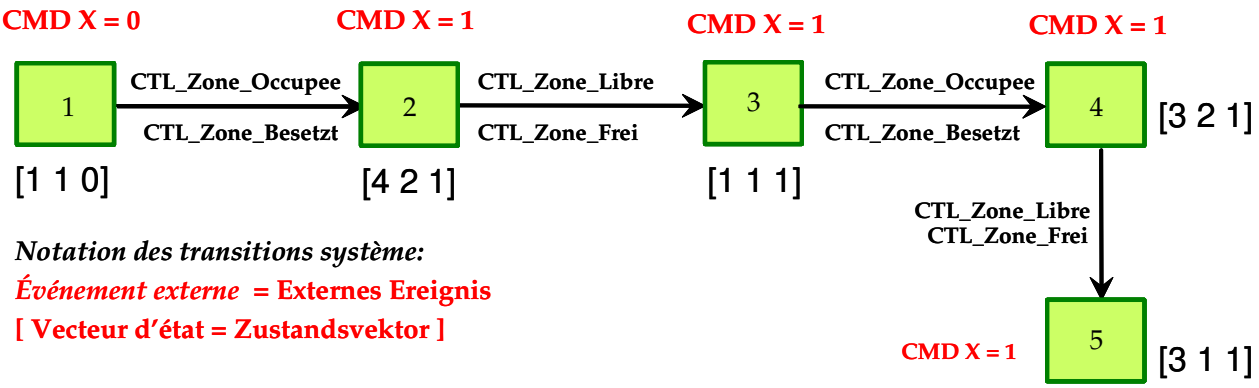



**Figure 6.22** : États systèmes avant et après une transition effectuée par le moteur du PIPC
**Abbildung 6.22**: System Zustände vor und nach einer Transition, die durch die Grundsoftware des PIPC durchgeführt wurde

Le respect du principe d'association d'un indicateur à une ou plusieurs places d'un unique graphe et le respect des conditions d'interprétation (automate déterministe à piles) permettent de répondre à nos attentes.

L'interprétation des graphes selon les méthodes des réseaux de Pétri (RdP) conduit à l'identification d'un nombre beaucoup plus important d'états système. Les états supplémentaires identifiés n'ont aucune réalité physique. Ils sont dus d'une part, à la non distinction des événements internes et externes et, d'autre part, au fait que le découpage en graphes élémentaires conduit à ce que plus d'une transition soit nécessaire entre deux états systèmes stables.

L'annexe D montre les limites classiques de l'interprétation des réseaux de Petri et l'exploration des états système. Sur des exemples nous ferons apparaître les écarts obtenus avec différentes options d'exploitation de l'outil ROMEO.

### 6.2.4.10 Règles générales

Afin de simplifier l'écriture des graphes de marquages, l'interprétation pratique des graphes ne dispose que d'un unique jeton :

- le marquage du graphe est alors le numéro de la place occupée (numéro compris entre 0 et 9). Le marquage des places n'est jamais utilisé en tant qu'Événement ou de Condition d'une transition ;
- les transitions partant d'une même place doivent toujours être exclusives, soit au moyen des événements, soit au moyen des conditions de réalisation de la transition.

Ces règles rendent l'interprétation des graphes déterministe.

Die Einhaltung des Grundsatzes der Verbindung eines Indikators mit einem oder mehreren Zuständen eines eindeutigen Graphen und die Einhaltung der Interpretationsbedingungen (deterministischer Stapelautomat) erlauben es, die an den Automaten gestellten Erwartungen zu erfüllen.

Die Interpretation der Graphen mithilfe der Petrinetzmethode führt zur Identifikation einer sehr großen Anzahl an Systemzuständen. Die zusätzlich identifizierten Zustände haben keinen physischen Gegenpart. Sie sind einerseits auf die mangelnde Differenzierung der internen und externen Ereignisse und andererseits auf die Teilung in elementare Graphen zurückzuführen. Letzteres führt dazu, dass mehr als ein Übergang zwischen zwei stabilen Systemzuständen notwendig ist.

Anhang D zeigt die bekannten Grenzen der Petrinetzinterpretation und der Auswertung der Systemzustände. Anhand von Beispielen werden Unterschiede aufgezeigt, die mit verschiedenen Optionen des ROMEO Anwendungsprogramms erhalten wurden.

### 6.2.4.10 Allgemeine Regeln

Um die Formulierung der Markierungsgraphen zu vereinfachen, verfügt die praktische Interpretation der Graphen nur über eine einzige Marke:

- die Markierung des Graphen ist die Nummer des besetzten Zustandes (Nummer zwischen 0 und 9). Die Markierung der Zustände wird niemals als Ereignis oder als Bedingung einer Transition benützt.
- die Transitionen, die vom selben Zustand ausgehen, müssen exklusiv sein, entweder durch die Ereignisse oder durch die Ausführungsbedingungen der Transition.

Diese Regeln führen dazu, dass die Interpretation der Graphen deterministisch ist.

### 6.2.4.11 Méthode de génération des états accessibles

Nous générons l'ensemble des états accessibles du système à partir de l'état suivant du système :

### 6.2.4.11 Methode der Erzeugung des erreichbaren Zustandraums

Alle erreichbaren Systemzustände werden ausgehend vom folgenden Zustand des Systems erzeugt:

$$E_i : [Marquage\ des\ Graphes; Etats\ des\ Indicateurs; Etats\ des\ Entrées] \quad \textbf{(12)}$$

$$E_i : [Markierung\ der\ Graphen; Zustände\ der\ Indikatoren; Zustände\ der\ Eingänge]$$

A partir d'un état $E_i$, il est possible de générer tous les états du système pouvant le suivre de la manière suivante :

- Pour chacun des automates du système, nous regardons quelles transitions peuvent être franchies. Il faut injecter chaque événement ;
- Pour qu'une transition puisse être franchie, il faut que l'état précédent de cette transition soit marqué et que l'indicateur (ou les autres conditions terrain) conditionnant éventuellement le franchissement de la transition soit (soient) bien positionné(s).

Ausgehend von einem Zustand $E_i$ werden alle Zustände des Systems erzeugt, die diesem folgendermaßen folgen können:

- Für jeden Automaten des Systems wird untersucht, welche Transitionen überschritten werden können. Hierbei muss man jedes Ereignis einlesen.
- Damit eine Transition geschaltet werden kann, muss der der Transition vorangehende Zustand markiert sein und der Indikator (oder andere Bedingungen vor Ort), der eventuell den Übergang beeinflusst, muss die richtige Position haben.

Remarque :

Si le franchissement d'une transition a pour action associée une activation interne (c'est à dire un message entre deux automates situés dans une même machine), nous traitons immédiatement cette activation et de manière instantanée au sens des états système. Donc, lors de l'envoi d'une activation interne, nous considérons que celle-ci est immédiatement traitée, le système est donc immédiatement dans l'état résultant de cette activation.

Nous ne tenons donc pas compte de l'état intermédiaire où cette activation est à traiter : si dans l'état $E_\alpha$, un message est traité, que le système passe dans l'état $E_\beta$, avec une activation interne à envoyer, celle-ci arrive et est traitée instantanément, et le système est alors dans un état $E_\gamma$.

Nous considérons que le système est passé directement de l'état $E_\alpha$ à l'état $E_\gamma$.

Cette façon d'effectuer la génération des états accessibles du système pose un problème, détaillé ci-après.

Bemerkung:

Wenn die Schaltung einer Transition eine interne Aktivierung auslöst (das heißt eine Kommunikation zwischen zwei Automaten der gleichen Maschine), wird diese Aktivierung sofort bearbeitet, wobei „sofort" im Sinne der Systemzustände, also ohne Zeitverlust, bedeutet. Die Sendung einer internen Aktivierung wird als sofort bearbeitet betrachtet. Das System ist also sofort in dem aus der Aktivierung entstandenen Zustand.

Der Zwischenzustand, in dem diese Aktivierung bearbeitet wird, wird also nicht berücksichtigt: wenn im Zustand $E_\alpha$ eine Nachricht verarbeitet wird, die besagt, dass das System in den Zustand $E_\beta$ übergeht und eine interne Aktivierung beendet werden soll, kommt diese sofort an und wird sofort behandelt. Das System ist dann also im Zustand $E_\gamma$. Man nimmt also an, dass das System direkt vom Zustand $E_\alpha$ in den Zustand $E_\gamma$ übergegangen ist.

Diese Art, erreichbare Systemzustände zu generieren, wirft jedoch ein Problem auf, das im Folgenden erläutert wird.

### 6.2.4.12 Problème posé par la génération des états accessibles

Lorsque l'interpréteur traite un événement externe, les actions sur les indicateurs sont placées dans la FIFO des événements internes. Lorsque l'interpréteur a fait évoluer le marquage des automates, et généré les actions externes, il positionne les indicateurs d'automates. Ensuite, il traite les messages internes les uns après les autres, suivant son algorithme de résolution.

Imaginons le cas suivant où les trois graphes X, Y et Z suivants sont interprétés sur la même machine (constitue un automate produit) :

### 6.2.4.12 Probleme der Zustandsraumerzeugung

Wenn der Interpreter ein externes Ereignis bearbeitet, werden die auf die Indikatoren bezogenen Handlungen im FIFO-Speicher für interne Ereignisse eingetragen. Sobald der Interpreter die Markierungen erneuert und die externen Handlungen erzeugt hat, werden die Indikatoren des Automaten umgestellt. Danach werden die internen Handlungen eine nach der anderen durch den Lösungsalgorithmus abgearbeitet. Abb. 6.35 stellt einen Fall dar, bei dem die drei Graphen X, Y und Z auf der gleichen Maschine interpretiert werden (einen Produktgraphen bilden).

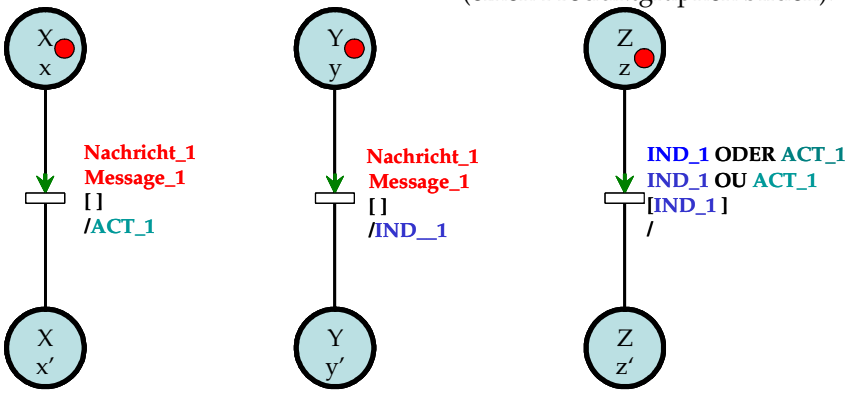



**Figure 6.35** : Génération des états accessibles / **Abbildung 6.35** : Erzeugung der zugänglichen Zustände

Trois graphes sont respectivement dans l'état x, dans l'état y et dans l'état z :

- Une transition relie l'état x à l'état x' du graphe X. Elle est déclenchée, sans condition, par l'occurrence de l'événement Message_1. L'action associée est une activation interne ACT_1 (message interne inter graphe).
- Une transition relie l'état y à l'état y' du graphe Y. Elle est déclenchée, sans condition, par l'occurrence du même événement Message_1. L'action associée est le positionnement à l'état «vrai» de l'indicateur IND_1 de l'automate global.
- Une transition relie l'état z à l'état z' du graphe Z. Elle est conditionnée par l'état «vrai» de l'indicateur IND_1. Elle est déclenchée par l'événement interne ACT_1 ou le passage à «vrai» de l'indicateur IND_1.
- L'occurrence de l'événement Message_1 fait franchir les transitions marquées des graphes X et Y, PUIS positionne à Vrai l'indicateur IND_1, PUIS l'interpréteur traite l'activation interne ACT_1 qui a été placée dans la FIFO interne. La transition marquée du graphe Z peut alors être franchie, puisque l'indicateur IND_1 est à «vrai»

Drei Graphen sind jeweils im Zustand x, y und z:

- Eine Transition verbindet den Zustand x und den Zustand x' des Graphen X. Er wird ohne zusätzliche Bedingung durch das Eintreffen der Nachricht 1 ausgelöst (interne Nachricht zwischen den Graphen).
- Eine Transition verbindet den Zustand y und den Zustand y' des Graphen Y. Er wird ohne zusätzliche Bedingung durch das Auftreten des gleichen Ereignisses „Nachricht 1" ausgelöst. Die dazugehörige Handlung ist das Umstellen des Indikators IND_1 des globalen Automaten auf die Position „wahr".
- Eine Transition verbindet den Zustand z mit dem Zustand z' des Graphen Z. Diese Transition ist bedingt durch die Position „wahr" des Indikators IND_1. Sie wird durch das interne Ereignis ACT_1 oder das Umstellen des Indikators IND_1 auf „wahr" schaltbar.
- Beim Auftreten des Ereignis „Nachricht 1" werden die markierten Transitionen der Graphen X und Y geschaltet, danach wird der Indikator IND_1 auf „wahr" umgestellt und danach bearbeitet der Interpreter die interne Aktivierung ACT_1, die im internen FIFO-Speicher zwischengespeichert wurde. Die markierte Transition des Graphen kann dann geschaltet werden, da der Indikator IND_1 „wahr" ist.

L'explorateur des états accessibles va traiter ce cas de la même manière suivante :

- Le traitement du Message_1 va d'abord faire évoluer un des deux graphes X et Y (selon l'ordre d'écriture dans le fichier texte).
- Si c'est le graphe X, l'activation interne ACT_1 va être générée et traitée immédiatement. L'indicateur interne IND_1 n'étant pas à 1 (vrai), la transition marquée du graphe Z ne pourrait être franchie si l'indicateur IND_1 n'était pas dans le champ événement. L'activation interne pourrait donc ne pas avoir d'effet dans ce cas.
- Si c'est le graphe Y, la survenue du Message_1 fera évoluer le graphe Y, l'indicateur IND_1 est alors positionné à «vrai». Lorsque le graphe X est activé, l'activation ACT_1 est générée et la transition marquée du graphe Z sera franchie.

Lors du traitement du même message, le réseau peut ne pas évoluer exactement donc pas de la même manière dans les deux cas. Néanmoins, le respect des règles d'interprétation permet d'obtenir dans les deux cas le même fonctionnement de l'automate, les mêmes états stables (Figure 6.36) :

Beim Auswerten der erreichbaren Zustände wird der vorliegende Fall auf folgende Art behandelt:

- Die Bearbeitung der Nachricht_1 wird zuerst auf einem der zwei Graphen X und Y durchgeführt (je nach der Reihenfolge in der Textdatei).
- Wenn es sich um den Graphen X handelt, wird die interne Aktivierung ACT_1 erzeugt und sofort bearbeitet. Da der interne Indikator IND_1 nicht auf 1 (wahr) steht, kann die markierte Transition des Graphen Z nur dann geschaltet werden, wenn sich der Indikator IND_1 nicht auf dem Feld „Ereignis" befindet. Die interne Aktivierung könnte also in diesem Fall unwirksam sein.
- Wenn der Graph Y als erstes eingelesen wird, verändert das Auftreten der Nachricht 1 als erstes Y; der Indikator IND_1 wird auf „wahr" umgestellt. Wenn der Graph X aktiviert ist, wird die Aktivierung ACT_1 erzeugt und die markierte Transition des Graphen Z geschaltet.

Beim Bearbeiten derselben Nachricht muss sich das Netz in beiden Fällen nicht unbedingt gleich verhalten. Aber das Einhalten der Interpretationsregeln erlaubt es, in beiden Fällen dieselbe Funktionsweise des Automaten zu erhalten und dieselben stabilen Zustände (Abb. 6.36).

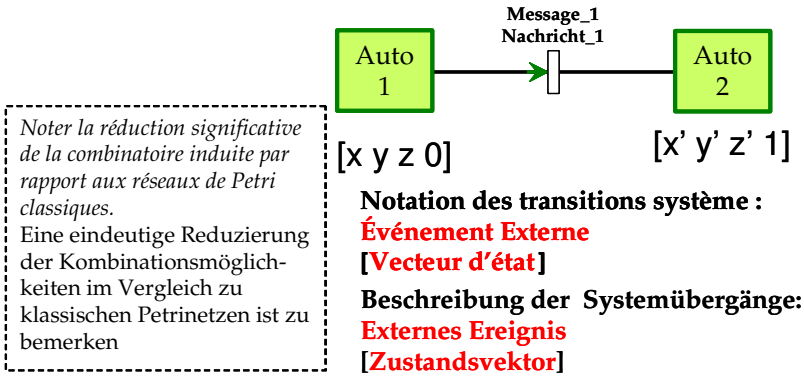



**Figure 6.36 :** Génération des états accessibles / **Abbildung 6.36**: Erzeugung der zugänglichen Zustände

### 6.2.4.13 Réduction de la combinatoire

Le nombre d'états accessibles du système est directement associé au nombre d'événements toujours occurrents et à la réceptivité des transitions. Le problème de l'explosion combinatoire est dû aux événements externes, à leurs combinaisons sans significations fonctionnelles.

Il est intéressant de noter la puissance de cette méthode en termes de réduction de la combinatoire lors de la génération de la liste des états accessibles du système. Le nombre théorique d'états accessibles est très supérieur au nombre d'états accessibles atteints par cette méthode et les hypothèses retenues.

### 6.2.4.13 Reduzierung der Kombinationen

Die Anzahl der zugänglichen Zustände des Systems ist direkt mit der Anzahl der immer wieder auftretenden Ereignisse und mit der Aufnahmefähigkeit der Transitionen verbunden. Das Problem der kombinatorischen Explosion ist den externen Ereignissen und ihren Kombinationen ohne funktionelle Bedeutung zuzuschreiben.

Die Stärke der hier vorgestellten Methode liegt in der Reduzierung der Kombinationen, vor allem bei der Erzeugung der Liste der erreichbaren Zustände. Die theoretische Anzahl der erreichbaren Zustände ist sehr viel höher als die unter den angenommen Hypothesen von dieser Methode erreichten Zustände.

La preuve s'applique sur un ensemble d'automates décrivant un processus fonctionnel. C'est cette notion qui définit le découpage des automates de l'application sur lesquels la preuve formelle va être réalisée.

S'il est possible d'identifier plusieurs processus fonctionnels indépendants (même s'ils contribuent tous les deux à une fonction Produit ou Somme), alors il est possible de réaliser autant de preuves formelles de spécifications indépendantes (réduction de la combinatoire).

## 6.2.5 Application du langage AEFD

La description du fonctionnement d'un poste d'aiguillage peut être exprimée sous forme de graphes d'état en interactions entre eux et avec l'environnement (entrées et sorties NS1). Ceux-ci peuvent indifféremment décrire le fonctionnement des différentes entités fonctionnelles ou reprendre les schémas de principe à relais d'un poste électrique existant. Illustrons l'utilisation pratique du langage AEFD pour décrire un fonctionnel ferroviaire par l'exemple de la Figure 6.23.

Der Beweis findet auf einer Anzahl von Automaten statt, die zusammen einen funktionellen Vorgang beschreiben. Und genau dieses Konzept bestimmt die Aufteilung der Automaten der Anwendung, auf denen der formale Beweis durchgeführt wird.

Wenn es möglich ist, mehrere unabhängige funktionelle Vorgänge zu identifizieren (selbst wenn beide zu einer Funktion „Produkt" oder „Summe" beitragen), kann man genauso viele unabhängige formale Beweise der Spezifikationen (Reduzierung der Kombinationen) durchführen.

## 6.2.5 Anwendung der AEFD-Sprache

Die Beschreibung der Funktionsweise eines Stellwerks kann in Form von untereinander und mit der Umwelt in Interaktion stehenden Graphen ausgedrückt werden (NS1 Ein- und Ausgänge). Diese können sowohl die Funktionsweise der verschiedenen funktionellen Einheiten, als auch die Prinzipschaltbilder eines bestehenden, elektrischen Relaisstellwerks beschreiben. Die praktische Nutzung der AEFD-Sprache für Eisenbahnfunktionen wird durch das Beispiel in Abb. 6.23 illustriert.

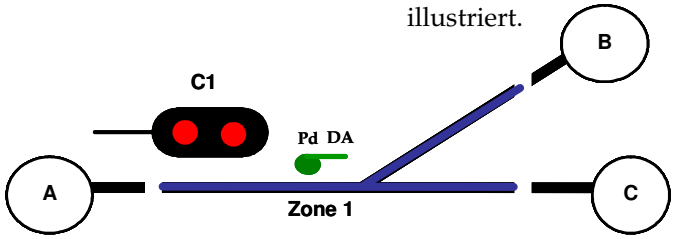



**Figure n°6.23** : Plan de voie de l'exemple d'application
**Abbildung 6.23**: Gleisplan des Anwendungsbeispiels

Il vient ainsi sur la Figure 6.24 pour la phase commande et préparation (Cf. Chapitre 4).

Für die Phase Vorbereitung (vgl. Kapitel 4) erhält man so das Schaubild aus Abb. 6.24.

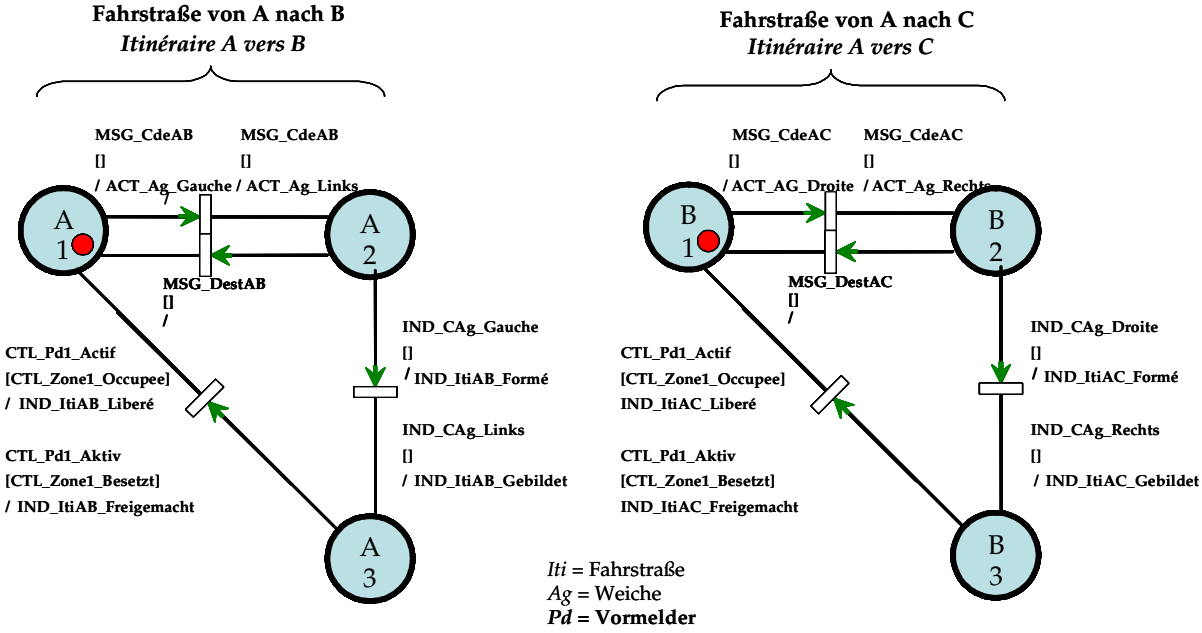



**Figure 6.24** : Graphes des phases commande et préparation de l'exemple d'application
**Abbildung 6.24**: Graph der Phasen Vorbereitung des Anwendungsbeispiels

Nous obtenons ainsi de même pour la phase enclenchement et ressources partagées (cf. Chapitre 4) les graphes de la figure 6.25.

Ebenso erhält man für die Phasen Sicherung und Aufteilung der Ressourcen (vgl. Kapitel 4) die Graphen aus Abb. 6.25.

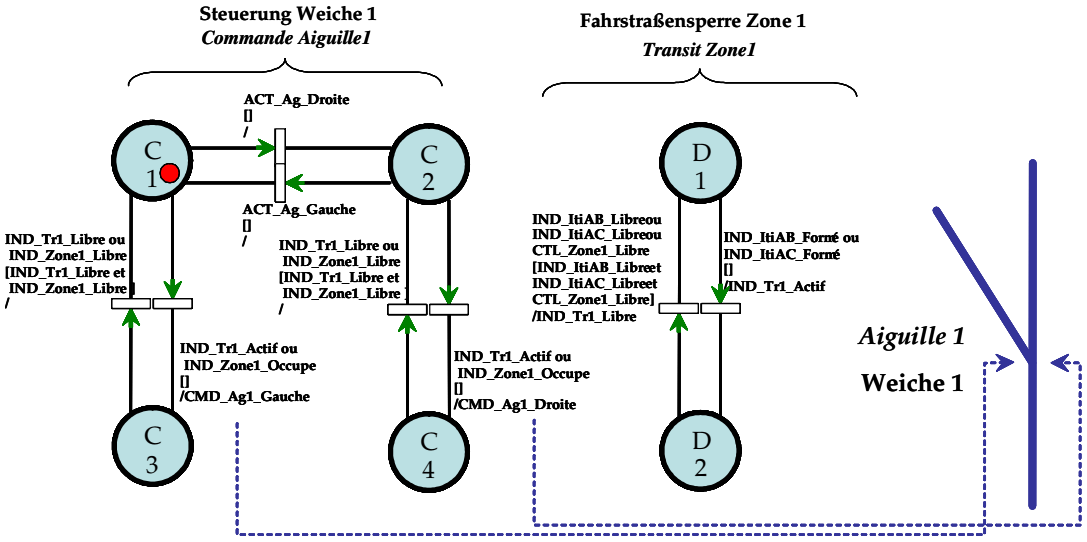



**Figure n°6.25** : Graphes des phases enclenchement et ressources de l'exemple d'application
**Abbildung n°6.25**: Graphen der Phasen Sicherung und Aufteilung der Ressourcen des Anwendungsbeispiels

La même démarche conduit à définir la phase de contrôle (établissement) de l'itinéraire (Cf. Chapitre 4). La Figure 6.26 illustre ceci.

Dasselbe Vorgehen führt dazu, die Phase der Fahrstraßenüberwachung zu definieren (Kapitel 4). Das Ergebnis ist in Abb. 6.26 dargestellt.

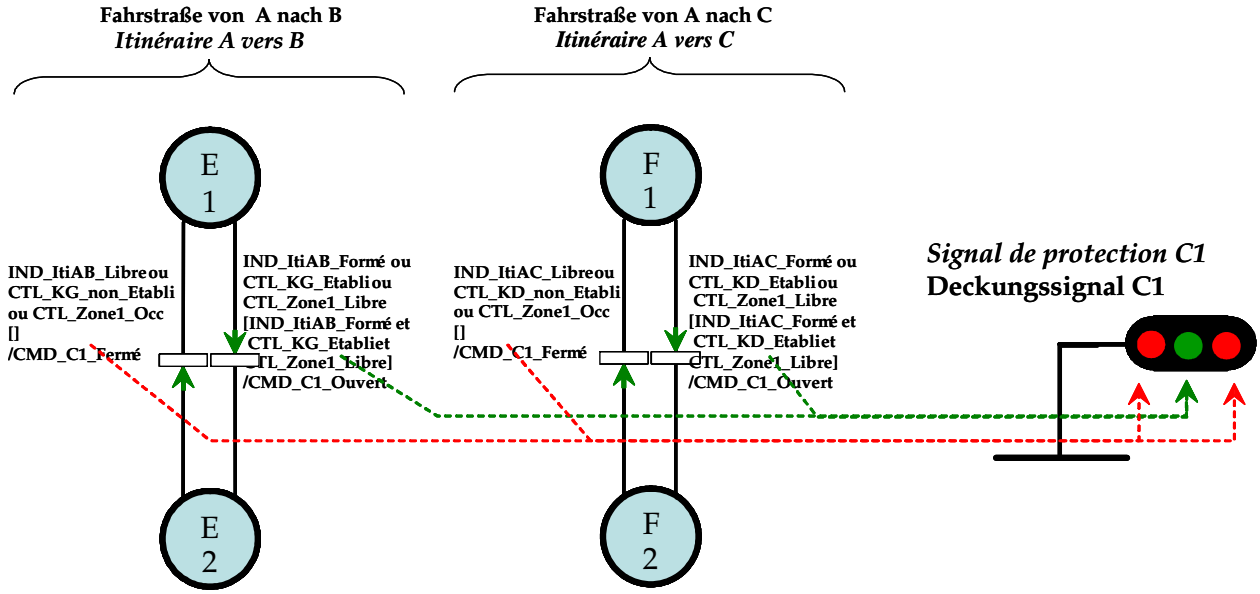



**Figure 6.26** : Graphes de la phase établissement de l'exemple d'application
**Abbildung 6.26**: Graphen der Phase Bildung des Anwendungsbeispiels

Les graphes décrivant chaque phase de fonctionnement sont en interaction. L'état courant du poste peut être résumé (est contenu) dans un vecteur d'état VE.

Il est à noter que le traitement d'un événement extérieur se traduit généralement par l'évolution de plusieurs marquages (plusieurs graphes de l'automate) et de plusieurs indicateurs.

Die Graphen, die jede Phase der Funktionen beschreiben, stehen untereinander in Wechselwirkung. Der vorliegende Zustand des Stellwerks kann in einem Zustandsvektor (VE) zusammengefasst werden (er ist in einem solchen Vektor enthalten).

Man muss feststellen, dass sich die Bearbeitung eines externen Ereignisses im Allgemeinen in der Änderung mehrerer Markierungen (mehrere Graphen des Automaten) und mehrerer Indikatoren äußert.

Ainsi le traitement du message de commande A-B conduit de l'état système [111111,110] à [312221,110]

So führt die Bearbeitung der Steuerungsmitteilung A-B vom Zustandsvektor [111111,110] zum Vektor [312221,110].

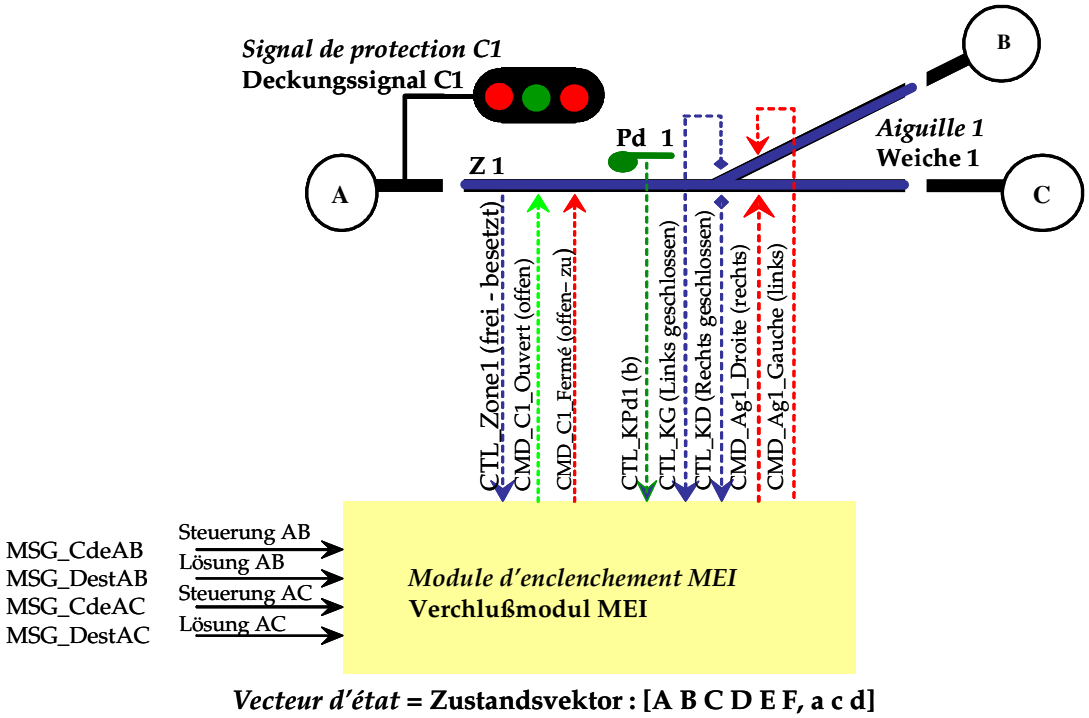



**Figure 6.27** : Module d'enclenchement et équipement terrain pour l'exemple d'application
**Abbildung 6.27**: Verschlussmodul und Anlage vor Ort des Anwendungsbeispiels

En tenant compte des règles d'interprétation propres au langage AEFD, l'exploration des états atteignables du système est toujours possible et de taille raisonnable en regard d'un modélisation directe en réseaux de Petri classiques (mettre que c'est pour la propagation interne dans les graphes avant toute prise en compte d'une nouvelle entrée terrain). Sur le cas précédent, dans l'optique d'une commande de l'itinéraire A vers B, nous obtenons le graphe partiel suivant des états accessibles :

Il est ainsi possible d'explorer à partir d'un état initial connu du système (par la méthode du pivot par exemple) tous les états réellement fonctionnellement accessibles du système (du logiciel d'application), de vérifier que ce nombre est fini et même de taille raisonnable. Nous obtenons bien ainsi la réalisation d'un automate à nombre fini d'états.

Le cas précédent sera utilisé pour être validé formellement au moyen de notre méthode et par notre outil de preuve.

Aufgrund der Interpretationsregeln, die der AEFD-Sprache zu Eigen sind, ist die Auswertung der erreichbaren Zustände des Systems immer möglich und von mäßigem Aufwand verglichen mit einer direkten Modellierung mit klassischen Petrinetzen. Bezüglich des vorherigen Falls und für eine Steuerung der Fahrstraße von A nach B erhält man den im Schaubild 6.28 dargestellten Teilbaum erreichbarer Zustände.

Es ist also möglich, ausgehend von einem bekannten Anfangszustand des Systems (zum Beispiel mit Hilfe der Methode des Angelpunktes), alle wirklich funktionell zugänglichen Systemzustände (Zustände der Anwendersoftware) auszuwerten und zu prüfen, dass die Anzahl der Zustände endlich ist und eine vernünftige Größe hat. Man erhält so als Umsetzung einen endlichen Automaten.

Das vorhergehende Beispiel wird nun benutzt, um die formale Validierung mit der neuen Methode und dem dazugehörigen Beweiswerkzeug darzustellen.

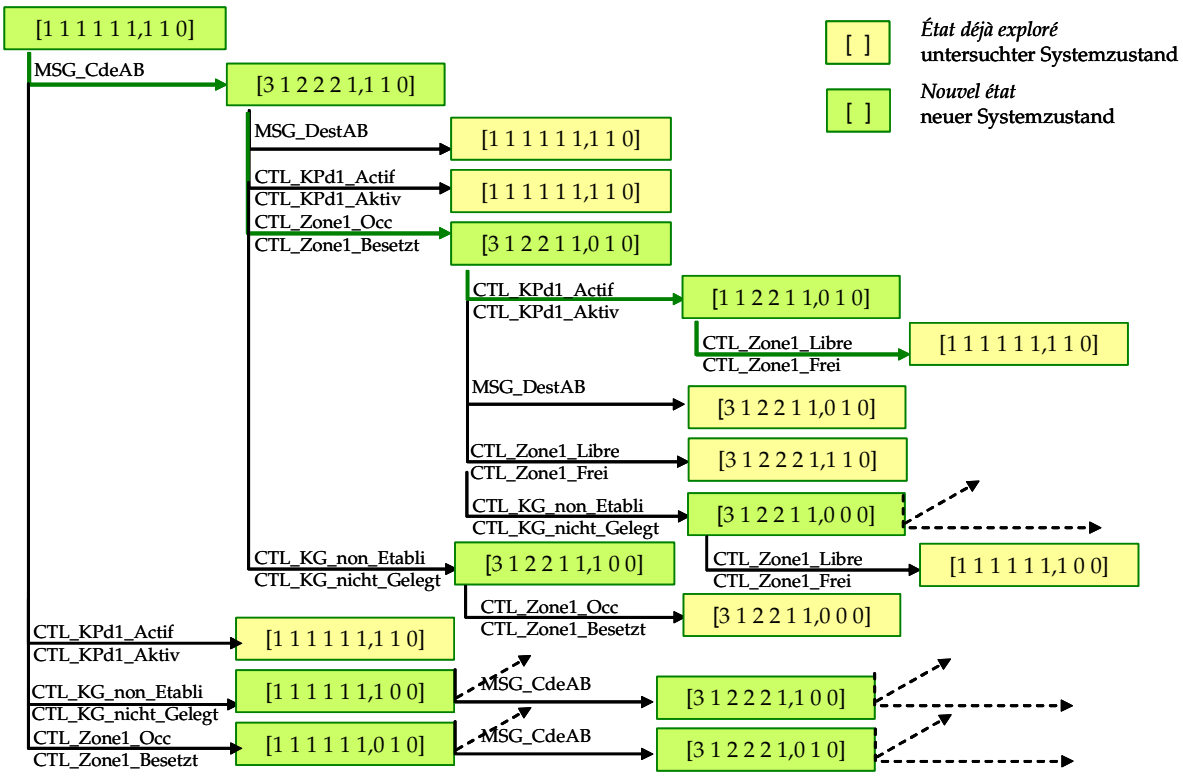



**Figure 6.28** : Arbre des états accessibles du système (avec en jaune les états déjà explorés)
**Abbildung 6.28**: Baum der erreichbaren Systemzustände (in gelb die schon ausgewerteten Zustände)

Notons que ce mode de description du fonctionnel applicatif (logiciel d'application) présente d'autres avantages pratiques :

- il est possible de recharger l'état courant du poste dans un état qui n'est pas l'état initial, en rechargeant simplement les composantes du vecteur d'état du poste. Cette possibilité a été utilisée dès 1995 pour le fonctionnel du Système Modulaire des Lignes (SYMEL) [Antoni, 2005] [Antoni, 2009] afin de gérer au mieux les modes dégradés et plus particulièrement les conditions de reprise du service après remise en état ;
- il est possible de détecter à des fins de maintenance et de surveillance tout écart par rapport au fonctionnement normal du système (le vecteur d'état courant du poste appartient toujours à un ensemble fini prédéterminé) ;
- il est possible de vérifier si chaque transition ainsi sollicitée respecte des conditions reposant sur des combinaisons d'indicateurs (IND) et/ou d'entrées terrain (CTL).

Nous utiliserons ces constats dans notre méthode de preuve.

Die vorgestellte Schreibweise der funktionellen Anwendung (Anwendersoftware) hat auch andere praktische Vorteile:

- Es ist möglich, die Stellwerke in einen Zustand zu versetzen, der nicht der Anfangszustand ist, indem man einfach die Komponenten des Zustandsvektors des Stellwerks einliest. Diese Möglichkeit wird seit 1995 vom SYMEL System benutzt [Antoni, 2005] [Antoni, 2009-4], um die Rückfallebene und die Wiederaufnahmebedingungen des Betriebs nach einer Instandhaltung bestmöglich zu verwalten.
- Es ist möglich bei der Wartung und bei der Überwachung jede Abweichung vom normalen Funktionieren des Systems festzustellen (der aktuelle Zustandsvektor des Stellwerkes gehört immer zu einer vorausbestimmten, endlichen Menge).
- Es ist möglich zu prüfen, ob jede auf diese Weise beanspruchte Transition die Bedingungen einhält, die auf Kombinationen von Indikatoren (IND) und/oder Informationen vor Ort (CTL) beruhen.

Diese Feststellungen werden in der neuen Beweismethode benützt.

## 6.3 La méthode de preuve retenue

### 6.3.1 Les automates à contraintes et preuve - Aspects Mathématiques

Les automates concurrentiels à contraintes sont des réseaux de PETRI particuliers. Les automates à contraintes décrivent des «automates à nombre fini d'états» ainsi que les graphes écrits à l'aide du langage AEFD. Les automates à contraintes [Narboni, 2001] sont des automates où l'on peut associer pour chaque transition une expression générique du type :

## 6.3 Die gewählte Beweismethode

### 6.3.1 Beschränkte Automaten und Beweis - Mathematische Aspekte

Konkurrierende Automaten mit Randbedingungen sind spezielle Petrinetze. Die Automaten mit Randbedingung beschreiben „endliche Automaten" sowie mittels der AEFD-Sprache geschriebene Graphen. Bei Automaten mit Randbedingungen [Narboni, 2001] kann jeder Übergang mit einer Formel folgenden Typs verbunden werden.

$$VE\left[i^{ème}\ injection\right] \leftarrow VE\left[(i-1)^{ème}\ injection\right] \neg \mathrm{T}\left\langle événement \middle| condition \right\rangle$$
$$VE\left[i^{-te}\ Eingabe\right] \leftarrow VE\left[(i-1)^{-te}\ Eingabe\right] \neg \mathrm{T}\left\langle Ereignis \middle| Bedingung \right\rangle \qquad \textbf{(1)}$$

L'expression (1) peut être lue et comprise de trois manières :

- Connaissant l'état de l'automate après le $i^{ème}$ événement externe traité et la nature de la transition valide à traiter
  $\Rightarrow$ l'état de l'automate au $(i\text{-}1)^{ème}$ événement externe traité est définie logiquement ;
- Connaissant l'état de l'automate après le $(i\text{-}1)^{ème}$ événement externe traité et la transition valide à traitée
  $\Rightarrow$ l'état de l'automate au $i^{ème}$ événement externe traité est définie logiquement ;
- Connaissant l'état de l'automate après le $i^{ème}$ événement externe traité et celui précédent (au $(i\text{-}1)^{ème}$ événement externe traité)
  $\Rightarrow$ les conditions de la transition sont définies logiquement.

Faire une preuve formelle de ce type d'automates revient à vérifier pour chaque état atteint une liste de propriétés logiques et ainsi répondre à la question : La conception respecte-t-elle les spécifications ?

La preuve repose sur la détection d'état puits, l'utilisation de mutuelles exclusions et de propriétés d'Invariance.

La preuve repose sur analyse des états accessibles du système et peut se résumer par l'expression :

$$Post^{*}\left(Init\right) \cap NonSûr = \phi\,? \qquad \textbf{(2)}$$

Die Formel (1) kann auf dreierlei Weise gelesen und verstanden werden:

- Ist der Zustand des Automaten nach dem $i$-ten bearbeiteten, externen Ereignis und die Natur der validierten Transition bekannt
  $\Rightarrow$ so ist der Zustand des Automaten beim Bearbeiteten des $(i\text{-}1)$-ten externen Ereignisses logisch definiert.
- Ist der Zustand des Automaten nach dem (i-1)-ten bearbeiteten, externen Ereignis und die Natur der validierten Transition bekannt,
  $\Rightarrow$ so ist der Zustand des Automaten beim Bearbeiten des $i$-ten externen Ereignisses logisch definiert.
- Ist der Zustand des Automaten nach dem $i$-ten und nach dem vorherigen bearbeiteten, externen Ereignis (also nach dem $(i\text{-}1)$-ten bearbeiteten, externen Ereignis) bekannt
  $\Rightarrow$ so sind die Bedingungen der Transition logisch definiert.

Ein formaler Beweis eines solchen Automatentyps läuft darauf hinaus, für jeden erreichten Zustand eine Liste logischer Eigenschaften zu prüfen und so die Frage zu beantworten: Respektiert die Konzeption die Spezifikationen? Der Beweis beruht auf der Suche absorbierender Zustände, der Benutzung von gegenseitigen Ausschlüssen und von Invarianzeigenschaften.

Der Beweis beruht auf der Analyse der zugänglichen Zustände des Systems und kann durch den folgenden Ausdruck zusammengefasst werden:

$$Nachher^{*}\left(Initial\right) \cap Unsicher = \phi\,? \qquad \textbf{(2)}$$

L'expression (2) traduit le fait que l'intersection entre l'ensemble des états accessibles du système et l'ensemble des états redoutés (non sûrs) est un ensemble vide.

Alors la preuve que le système ne peut atteindre un des états non sûr est assurée. La preuve est rejetée dans le cas contraire.[71]

## 6.3.2 La méthode de preuve

Notre méthode est une méthode de validation formelle pour les propriétés de sécurité (direct et indirect par les procédures et les opérateurs). Elle couvre tous les champs de la spécification jusqu'à l'exécution. Notre démarche s'approche d'une démarche de Model Checking qui s'appliquerait non pas sur un «modèle» mais sur le code cible exécutable (interprété). Nous n'exécutons pas de scripts d'essai mais nous vérifions complètement que le système suit les propriétés de sécurité attendues à chaque moment. Elle est automatisée et l'application n'exige pas d'interventions manuelles.

Le principe de la méthode est donc simple. L'idée fondamentale est : « Si une propriété est vraie dans un état (marqué) occupé et si la conservation de cette propriété pendant la transition qui suit cet état est garantie la propriété sera vraie dans le nouvel état occupé. La démonstration peut être continuée aussi longtemps que la propriété est préservée » (Figure 6.29).

Ce principe est utilisé pour les propriétés de sécurité et les propriétés fonctionnelles.

Der Term (2) übersetzt die Tatsache, dass die Schnittmenge der zugänglichen Zustände des Systems und der Menge der befürchteten Zustände (unsicher) eine leere Menge sein soll.

Somit ist der Nachweis, dass das System keine unsicheren Zustände erreichen kann, erbracht. Der Beweis wird im gegenteiligen Fall abgelehnt[72].

## 6.3.2 Überprüfungsmethode

Die vorgestellte Methode ist eine Methode der formalen Überprüfung der Sicherheitseigenschaften (direkte und indirekte die Vorschriften und das Bedienungspersonal betreffend). Die Methode deckt alle Bereiche von der Spezifikation bis hin zur Ausführung ab. Das vorgestellte Vorgehen ähnelt einem „Model Checking"-Ansatz, der jedoch nicht auf einem „Modell", sondern auf dem ausführbaren (interpretierten) Zielcode ausgeführt wird. Es werden keine Versuchsskripte ausgeführt, sondern es wird vollständig nachgewiesen, dass das System zu jedem Zeitpunkt den erwarteten Sicherheitseigenschaften genügt. Die Methode ist automatisiert, und ihre Anwendung erfordert keine manuellen Eingriffe.

Das Prinzip der Methode ist also einfach. Der Grundgedanke ist: Wenn eine Eigenschaft in einem belegten (markierten) Zustand wahr ist, und wenn die Erhaltung dieser Eigenschaft während der Transition während der diesem Zustand folgenden Transition, garantiert ist, ist die Eigenschaft auch im neuen Zustand wahr. Dieser Beweis kann fortgesetzt werden, solange die Eigenschaft beibehalten wird (Abb. 6.29).

Dieses Prinzip wird sowohl für Sicherheitseigenschaften, als auch für die funktionellen Eigenschaften verwendet.

[71] Il est à noter que c'était déjà le principe de la démonstration de sécurité absolue de la méthode Descubes [Descubes, 1898]

[72] Dieser Begriff war schon so benützt bei der Descubes Methode [Descubes, 1898]

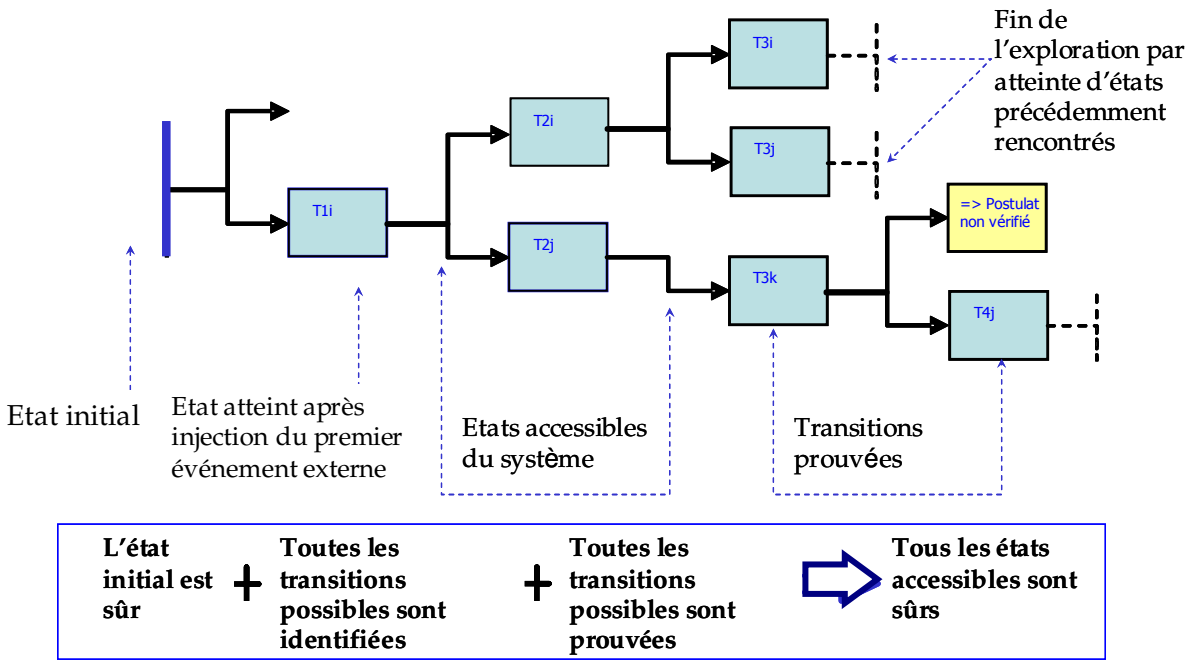



**Figure 6.29** : Exploration et preuve formelle des propriétés de sécurité formalisées sous forme d'AP (Automate de Preuve) au fur et à mesure de l'exploration des états accessibles de l'automate

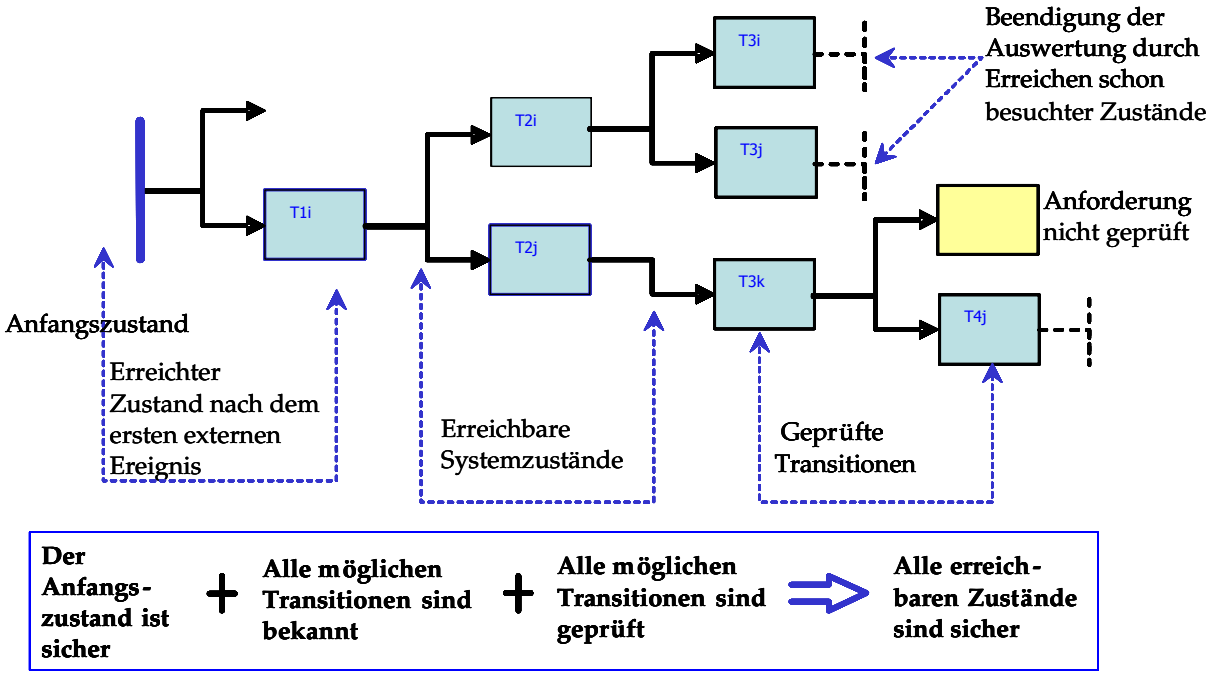



**Abbildung 6.29**: Auswertung und formaler Beweis der Sicherheitseigenschaften, die in Form von AP (Beweisautomaten) formalisiert wurden, je nach Stand der Auswertung der erreichbaren Zustände des Automaten

De nombreux travaux ont été menés sur les automates à nombre fini d'état. Notre méthode, retenue pour sa simplicité et sa possibilité d'automatisation, a été inspirée de celle proposée par Bielinski en 1993 [Bielinski, 1993] dans le cadre de ses travaux sur la validation de circuits intégrés d'encodage et de décodage de code CRC.

Le fonctionnel applicatif du poste d'aiguillage à prouver (ou de l'automatisme le cas échéant) est spécifié et réalisé sous la forme d'un automate à états finis avec :

- E1 un vecteur des entrées,
- S1 un vecteur des sorties sans erreur,
- S2 un vecteur des sorties avec erreur,
- P3 un vecteur d'état du prouveur.

Prouver l'automate, c'est montrer que s'il existe une erreur de traitement alors :

$$\forall(E1, S2) \Rightarrow P3 \neq OK \qquad \textbf{(3)}$$

La méthode utilisée s'inspire de celle des invariants en la transposant au cas des automates :

- un invariant est attaché à chaque état de l'automate (ou du moins à chaque état d'une fonctionnalité à prouver) ;
- les invariants sont directement inspirés de l'écriture des incompatibilités entre positions de graphes du poste d'aiguillage, plus précisément des enclenchements correspondants. Les états accessibles du système (automate fonctionnel) sont identifiés par une méthode classique d'exploration d'états.

La démonstration repose sur une analyse d'accessibilité des états du système (automate global).

## *Exemple d'automate à états finis*

Illustrons ces concepts sur un exemple simple. Il s'agit d'un graphe chargé de calculer la somme S telle que :

$$S = \sum_{i=0}^{j} i \qquad \textbf{(4)}$$

Cette somme peut aussi s'écrire sous la forme d'un automate à états (Figure 6.30).

Zahlreiche Arbeiten haben sich mit endlichen Automaten beschäftigt. Die hier vorgestellte Methode, die aufgrund ihrer Einfachheit und ihrer Automatisierungsmöglichkeit ausgewählt wurde, ist von einer Methode inspiriert, die Bielinski [Bielinski, 1993] im Jahre 1993 im Rahmen seiner Arbeiten über die Validierung integrierter Schaltkreise für die Codierung und die Decodierung von CRC-Code vorgeschlagen hat.

Die funktionelle Anwendung des zu prüfenden Stellwerks (oder auch der Steuerung) wird spezifiziert und in Form eines endlichen Automaten dargestellt:

- E1: Eingangsvektor
- S1: fehlerfreier Ausgangsvektor
- S2: fehlerbehafteter Ausgangsvektor
- P3: Zustandsvektor des Prüfers.

Den Automaten zu prüfen heißt nachzuweisen, dass bei einem Bearbeitungsfehler gilt:

$$\forall(E1, S2) \Rightarrow P3 \neq OK \qquad \textbf{(3)}$$

Die benutzte Methode ist an die Invariantenmethode angelehnt und wurde für Automaten abgewandelt:

- Eine Invariante ist mit jedem Zustand des Automaten verbunden (oder wenigstens mit allen zu den prüfenden Funktionen gehörenden Zuständen).
- Die Invarianten sind direkt von der Formulierung der Widersprüche zwischen den Graphenpositionen eines Stellwerks abgeleitet, genauer gesagt von den dazugehörigen Verschlüssen. Die erreichbaren Zustände des Systems (funktioneller Automat) wurden durch eine herkömmliche Methode der Zustandsauswertung identifiziert.

Der Beweis beruht auf einer Analyse der erreichbaren Systemzustände (globaler Automat).

## *Beispiel des endlichen Automaten*

Dieses Konzept wird anhand eines einfachen Beispiels illustriert. Es handelt sich um einen Graphen, der die Summe S berechnen soll:

$$S = \sum_{i=0}^{j} i \qquad \textbf{(4)}$$

Diese Summe lässt auch in Form eines Zustandsautomaten ausdrücken (siehe Abb. 6.30).

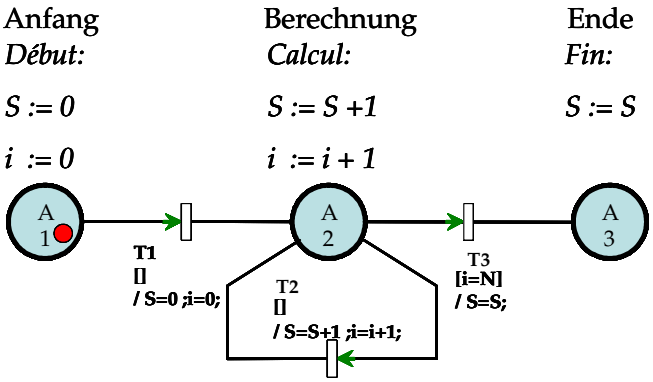



**Figure 6.30** : Exemple fonctionnel simple à prouver
**Abbildung 6.30** : Einfach zu beweisendes funktionelles Beispiel

Automate simple:

$$\begin{aligned} &T1 : Début \rightarrow Calcul; T1; i \leftarrow 0 \; et \; S \leftarrow 0 \\ &T2 : Calcul \rightarrow Calcul; i \leq N; i \leftarrow i+1 \\ &\qquad et \; S \leftarrow S+1 \\ &T3 : Calcul \rightarrow Fin; i = N; i = N \; et \; S \leftarrow S \end{aligned} \quad (5)$$

Invariants :

$$\begin{aligned} &Début : \phi \\ &Calcul : S = \sum_{j=0}^{i} j \\ &Fin : S = \frac{N.(N+1)}{2} \end{aligned} \quad (6)$$

Pour prouver cet automate, il faut montrer que :

- l'invariant *Début* est Vrai
- les transitions permettent bien de passer d'un Invariant au suivant jusqu'à l'état final.

*Début* est forcément vrai : l'invariant vide (vide à la mise sous tension)

$$Début \rightarrow Calcul : \quad (7)$$

$$i = 0 \text{ et } S = 0 \text{ et } S = \sum_{j=0}^{i} j.$$

$$Calcul \rightarrow Calcul : \quad (8)$$

$$S = \sum_{j=0}^{i} j \text{ et } i' = i+1 \text{ et } S' = S+1$$

$$\Rightarrow S' = i + \sum_{j=0}^{i} j = \sum_{j=0}^{i+1} j = \sum_{j=0}^{i'} j$$

$$Calcul \rightarrow Fin : \quad (9)$$

$$S = \sum_{j=0}^{i} j \; et \; i = n \; et \; S' = S$$

$$\Rightarrow S' = S = \sum_{j=0}^{i} j = \sum_{j=0}^{n} j = \frac{n.(n+1)}{2}.$$

Einfacher Automat:

$$\begin{aligned} &T1 : Anfang \rightarrow Berechnung; T1; i \leftarrow 0 \; et \; S \leftarrow 0 \\ &T2 : Berechnung \rightarrow Berechnung; i \leq N; i \leftarrow i+1 \\ &\qquad und \; S \leftarrow S+1 \\ &T3 : Berechnung \rightarrow Ende; i = N; i = N \; et \; S \leftarrow S \end{aligned} \quad (5)$$

Invarianten:

$$\begin{aligned} &Anfang : \phi \\ &Berechnung : S = \sum_{j=0}^{i} j \\ &Ende : S = \frac{N.(N+1)}{2} \end{aligned} \quad (6)$$

Um diesen Automaten zu prüfen muss man zeigen, dass:

- die Invariante *Anfang* wahr ist,
- die Transitionen erlauben es den Invarianten, zum nächsten Zustand zu gelangen bis hin zum Endzustand.

*Anfang* ist notgedrungen wahr: die leere Invariante (leer bei Inbetriebnahme)

$$Anfang \rightarrow Berechnung : \quad (7)$$

$$i = 0 \text{ und } S = 0 \text{ und } S = \sum_{j=0}^{i} j.$$

$$Berechnung \rightarrow Berechnung : \quad (8)$$

$$S = \sum_{j=0}^{i} j \text{ und } i' = i+1 \text{ und } S' = S+1$$

$$\Rightarrow S' = i + \sum_{j=0}^{i} j = \sum_{j=0}^{i'} j.$$

$$Berechnung \rightarrow Ende : \quad (9)$$

$$S = \sum_{j=0}^{i} j \text{ und } i = n \text{ und } S' = S$$

$$\Rightarrow S' = S = \sum_{j=0}^{i} j = \sum_{j=0}^{n} j = \frac{n \cdot (n+1)}{2}.$$

Cette démonstration est volontairement simplifiée, elle n'est toutefois pas complètement réaliste. En effet, l'automate présenté n'est pas complet, il suppose que les autres conditions ne se produisent jamais.

Néanmoins une difficulté apparaît, il faut définir un invariant pour chaque état de l'automate, ce qui demande une intervention humaine, *a priori* non automatisable.

## 6.3.3 Application méthodologique

### 6.3.3.1 Général

Notre méthode s'appuie sur les propriétés mathématiques des graphes d'état et sur une méthode de preuve (preuve par assertion) [Bielinski, 1993] [Narboni, 2001]. Elle s'applique à des graphes d'état comportant un seul jeton, dont les transitions sont instantanées, les évènements doivent être pris en en compte de manière séquentielle.

La méthode disant que « si une propriété est vraie dans une place occupée (marquée) et s'il existe un lemme (proposition vraie) qui montre la conservation de cette propriété au cours de la transition qui suit cette place alors la propriété sera vraie dans la nouvelle place occupée. On peut poursuivre la démonstration de proche en proche tant que la propriété est conservée » a pu être appliquée avec succès [Bielinski, 93].

La difficulté de la méthode est en fait de trouver et de démontrer de tels lemmes qui permettent d'évaluer la propriété dont on cherche à prouver la conservation pour chacune des transitions à franchir.

La méthode peut donc bien être appliquée à des fonctions d'enclenchement ferroviaire spécifiées sous forme de graphes interprétables par les postes PIPC. Néanmoins son application ne peut être automatisée en l'état : l'écriture des propriétés de sécurité et des lemmes nécessaires à la preuve est manuelle.

Nous devons donc trouver une avancée pour automatiser d'une part, l'écriture des propriétés et des lemmes et d'autre part, l'application de la méthode.

Il s'agit de vérifier qu'un automate (ensemble de graphes) vérifie en permanence une ou plusieurs propriétés : ces propriétés sont décrites par des graphes particuliers dit « automates de preuve ».

Pour faire la preuve d'un automate, il faut être capable de :

Dieser Beweis wurde extra vereinfacht und ist nicht unbedingt völlig realistisch. In der Tat ist der vorgestellte Automat nicht vollständig, es wurde angenommen, dass nie andere Bedingungen eintreten können.

Trotzdem tritt hier eine Schwierigkeit auf; man muss für jeden Zustand des Automaten eine Invariante definieren. Hierzu ist ein menschlicher Eingriff nötig, der *a priori* nicht automatisiert werden kann.

## 6.3.3 Methodischer Ansatz

### 6.3.3.1 Allgemein

Die vorgestellte Methode stützt sich auf die mathematischen Eigenschaften der Zustandsgraphen und auf eine Beweismethode (Beweis durch Analyse der Behauptung) [Bielinski, 1993] [Narboni, 2001]. Sie findet bei Zustandsgraphen mit nur einer Markierung, augenblicklichen Transitionen und sequentieller Bearbeitung der Ereignisse Anwendung.

Die Methode besagt, dass „wenn eine Eigenschaft in einem besetzten (markierten) Zustand wahr ist, und wenn es ein Lemma (wahrer Satz) gibt, das die Erhaltung dieser Eigenschaft während der Transition in den folgenden Zustand garantiert, dann ist die Eigenschaft auch im neu besetzten Zustand wahr. Dieser Beweis kann weitergeführt werden solange die Eigenschaft beibehalten wird. Diese Methode konnte erfolgreich angewandt werden [Bielinski, 1993].

Die Schwierigkeit der Methode besteht in der Tat darin, Lemmata zu finden und zu beweisen, die es erlauben, die Eigenschaft auszuwerten, deren Beibehaltung für jede der zu schaltenden Transitionen bewiesen werden soll.

Die Methode kann also gut auf Funktionen von Eisenbahnsicherungen angewandt werden, die in Form von durch PIPC-Stellwerke interpretierbaren Graphen spezifiziert wurden. Trotzdem kann ihre Anwendung nicht im vorliegenden Zustand automatisiert werden.

Hier sind also noch Fortschritte zu machen, um einerseits die Formulierung der Eigenschaften und der Lemmata und anderseits die Anwendung der Methode zu automatisieren.

Es soll kontrolliert werden, dass ein Automat (eine Menge von Graphen) ständig eine oder mehrere Eigenschaften erfüllt: diese Eigenschaften werden durch besondere Graphen, den sogenannten „Beweisautomaten", beschrieben.

. Um den Beweis eines Automaten durchzuführen, muss man fähig sein:

- émuler son fonctionnement : automate fonctionnel et automate de preuve ;
- décrire le fonctionnement de son environnement (automate de postulats) ;
- explorer tous ses états accessibles ;
- vérifier que toutes les propriétés sont respectées dans chacun des états explorés.

On aboutit au schéma suivant, simplification de l'Annexe D:

- sein Funktionieren für den funktionellen Automaten und den Beweisautomaten zu emulieren.
- das Funktionieren seiner Umwelt (Anforderungsautomat) zu beschreiben.
- all seine erreichbaren Zustände auszuwerten
- zu prüfen, dass alle Eigenschaften in jedem der explorierten Zustände eingehalten werden.

Man gelangt so zum Schema der Abb. 6.31, Simplifizierung des Anhangs D.

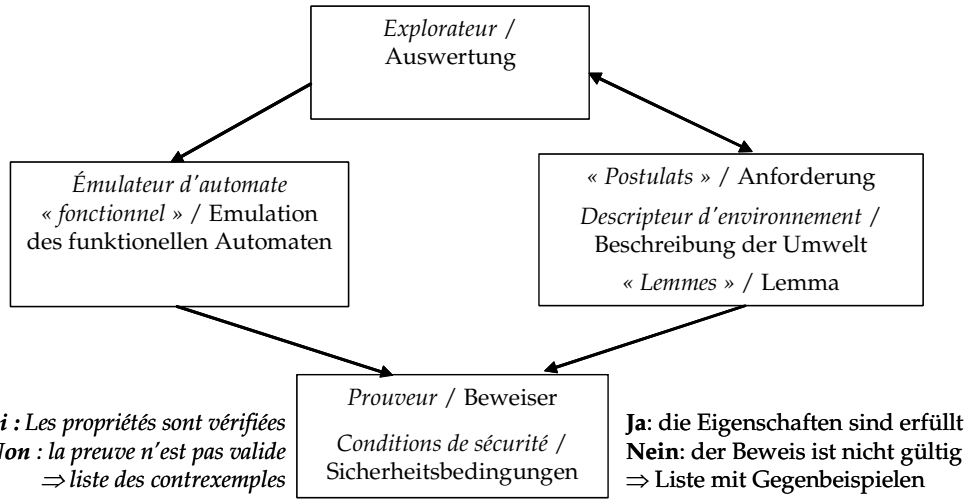



**Figure 6.31** : Principe de l'automatisation de la méthode de preuve
**Abbildung 6.31**: Prinzip der Automatisierung der formalen Methode

## 6.3.3.2 Étapes de la méthode

Il s'agit d'établir une preuve de non atteinte des événements redoutés (associés aux risques) à partir de la spécification d'un automate, étant à la fois la représentation (modélisation) de l'installation et l'application exécutée sur la machine cible. La démonstration repose sur l'étude des transitions des automates et se décompose en phases :

- L'énumération de tous les états et transitions possibles à partir de l'état initial. Il est à noter que l'énumération des états possibles du système n'est pas une simulation du système global. En effet on se place du point de vue de l'automatisme, et non de celui d'un observateur. Cette énumération est construite à partir de l'état initial (état à la mise en service du poste) en faisant une ramification sur les états accessibles. Cette méthode du point fixe donne également toutes les transitions possibles à partir de chaque état (arborescence) ;
- La mise en place de postulats et lemmes démontrant que si, dans un état, une propriété est vérifiée, alors, après le franchissement d'une transition possible, cette propriété reste vérifiée. Pour chaque transition un lemme est établi à l'aide d'hypothèses sur le fonctionnement du terrain ;

## 6.3.3.2 Methodische Phasen

Es soll der Beweis des Nichterreichens der befürchteten Ereignisse (in Bezug auf die damit verbundenen Risiken) erbracht werden, indem von der Spezifizierung eines Automaten ausgegangen wird, der sowohl die Anlage veranschaulicht (Modellierung), als auch die Anwendung auf der Zielmaschine darstellt. Der Beweis beruht auf der Analyse der Transitionen der Automaten und kann in folgende Phasen aufgeteilt werden:

- Die Aufzählung aller möglichen Zustände und Transitionen, vom Anfangszustand ausgehend. Man beachte, dass die Aufzählung der möglichen Systemzustände keine Simulation des globalen Systems ist. In der Tat wird das System von der Steuerung aus betrachtet und nicht aus der Sicht eines externen Beobachters, der auch die Zugbewegungen sehen würde. Diese Aufzählung wird vom Anfangszustand aus vorgenommen (Zustand bei Inbetriebnahme des Stellwerkes), indem man eine Verzweigung zu allen möglichen Zuständen erstellt. Diese Methode des Fixpunktes liefert zusätzlich für jeden Zustand alle möglichen Transitionen (Baumdiagramm).
- Die Anforderungen und Lemmata zeigen, dass eine Eigenschaft, wenn sie in einem Zustand wahr ist, auch nach einer möglichen Transition erhalten bleibt. Für jede Transition wird mithilfe der Hypothesen über das Funktionieren vor Ort ein Lemma aufgestellt

- Une récurrence sur le temps. Enfin, la démonstration utilise une récurrence sur le temps: dans l'état initial E0, les propriétés sont vérifiées, les lemmes et postulats ont démontré que si les propriétés sont vérifiées avant une transition possible, les propriétés restent vérifiées après, donc, quelque soit la suite de transitions (donc le temps), les propriétés sont vérifiées ;
- Nous avons cherché une solution permettant, de manière automatique, de démontrer les lemmes de façon aisée et rapide.

C'est pourquoi nous avons construit des automates de preuve. Ceux-ci évoluent en même temps que l'on fait la génération des états accessibles du système, et traduisent les hypothèses lors du passage entre deux états du système, ou lorsque le système reste dans un état donné.

La difficulté majeure à lever réside dans la réalisation d'une exploration exhaustive des états systèmes réels, sans sombrer dans une explosion combinatoire. Là encore, c'est au moyen d'automates particuliers que nous allons décrire.

## 6.3.4 Les automates de preuve

### 6.3.4.1 Principe général

Les propriétés de sécurité à prouver doivent pouvoir être définies par des opérateurs Jamais ou Toujours appliqués sur les composantes du vecteur d'état courant de l'automate.

Lorsqu'une propriété est valable quelque soit l'état courant de l'automate (état système du poste), elle peut faire l'objet d'un automate de preuve (AP) à deux places. Lorsque la situation redoutée (interdite) vient à se produire l'automate doit prendre une place particulière et l'indicateur P_5 (Contraire à la Sécurité) est positionné à Vrai.

Lorsqu'une propriété ne doit être valable que lorsque l'état courant de l'automate appartient à un sous-ensemble des états accessibles de l'automate (états système du poste), elle peut faire l'objet d'un automate de preuve (AP) à plus de deux places. Lorsque l'état courant du système appartient au sous-ensemble cible, l'AP prend une place particulière activant la transition représentant la propriété à vérifier.

- Eine Induktion über die Zeit. Schließlich benutzt man für den Beweis eine Induktion über die Zeit. Im Anfangszustand E0 werden die Eigenschaften nachgewiesen. Die Lemmata und Anforderungen zeigen, dass, wenn die Eigenschaften vor einer möglichen Transition wahr sind, diese Eigenschaften auch nach der Transition wahr bleiben. D.h. in welcher Reihenfolge (zu welchem Zeitpunkt) auch immer die Transitionen stattfinden, die Eigenschaften sind wahr.
- Es wurde eine Lösung gesucht, welche die Lemma automatisch, einfach und schnell beweist.

Aus diesem Grund werden die Beweisautomaten konstruiert. Diese verändern sich gleichzeitig mit der Erzeugung der zugänglichen Systemzustände und setzen die Hypothesen bei einer Transition zwischen zwei Zuständen oder bei einem Verbleiben in einem bestimmten Zustand um.

Die Hauptschwierigkeit besteht in der Verwirklichung einer erschöpfenden Auswertung der Zustände des realen Systems, ohne Probleme mit einer kombinatorischen Explosion zu bekommen. Auch dieses Problem wird mithilfe spezieller Automaten gelöst, was im Folgenden beschrieben wird.

## 6.3.4 Beweisautomaten

### 6.3.4.1 Allgemeine Grundlage

Die zu beweisenden Sicherheitseigenschaften müssen durch die Operatoren „Nie“ und „Immer“ definiert werden können, die auf den aktuellen Zustandsvektor des Systems angewendet werden.

Wenn eine Eigenschaft ungeachtet des aktuellen Zustands des Automaten gültig ist (Zustände des Stellwerks), kann sie Gegenstand eines Beweisautomaten (AP) mit zwei Zuständen werden. Wenn die befürchtete (verbotene) Situation eingetreten ist, muss der Automat einen speziellen Zustand (AP) einnehmen und der P_5-Indikator (Eigenschaft in Widerspruch zur Sicherheit) wird auf „wahr“ umgestellt.

Wenn eine Eigenschaft nur gültig sein darf, wenn der aktuelle Zustand des Automaten zu einer Teilmenge der zugänglichen Zustände des Automaten gehört (Systemzustände des Stellwerkes), kann sie Gegenstand eines Beweisautomaten (AP) mit mehr als zwei Zuständen werden. Wenn der aktuelle Systemzustand zur Zielteilmenge gehört, nimmt der AP einen speziellen Zustand ein, der die zu prüfende Eigenschaft darstellende Transition schaltbar macht.

A ce moment, lorsque la situation redoutée (interdite) vient à se produire l'automate de preuve (cf. figure 6.32) doit prendre une place particulière (AP[9] et l'indicateur *P_5* (propriété Contraire à la Sécurité) est positionné à Vrai.

In dem Moment, in dem die befürchtete (verbotene) Situation eintritt, muss der Automat (Abb. 6.32) einen speziellen Zustand einnehmen (AP[9]) und der *P_5*-Indikator (Eigenschaft in Widerspruch zur Sicherheit) wird auf „wahr" umgestellt.

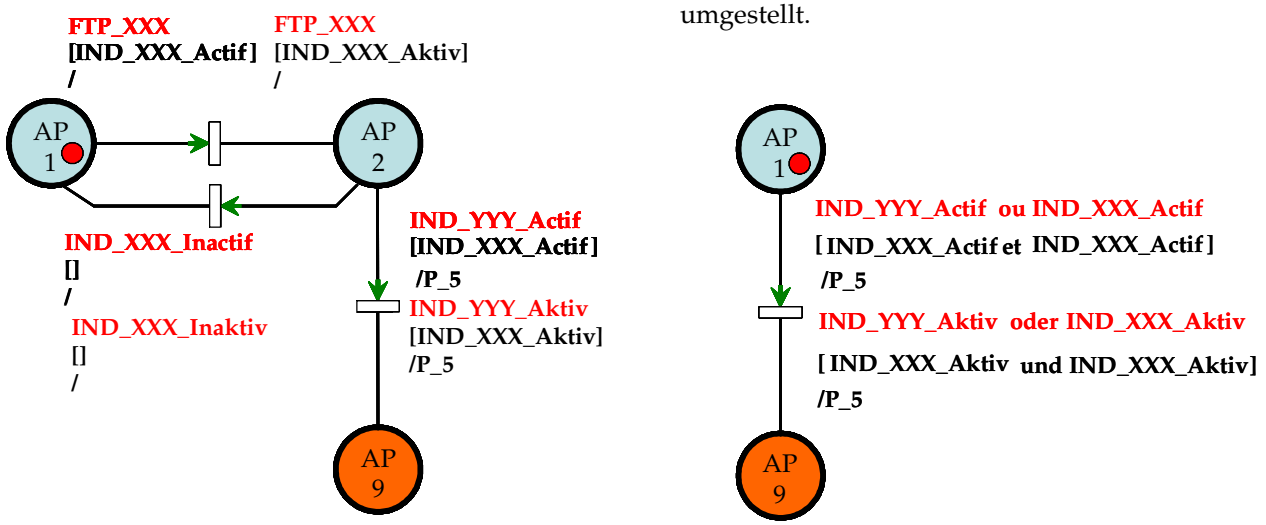



**Figure 6.32** : Automates de preuve pour la vérification d'une propriété de sécurité soit un sous ensemble des états accessibles de l'automate, soit pour l'ensemble de ces mêmes états
**Abbildung 6.32**: Beweisautomaten für die Prüfung einer Sicherheitseigenschaft entweder auf eine Teilmenge der zugänglichen Zustände des Automaten, oder für die Gesamtheit dieser Zustände

Dans le premier cas : La propriété de sécurité à vérifier s'exprime par l'incompatibilité entre les indicateurs internes *IND_YYY_Actif* et *IND_XXX_Actif*. Les deux indicateurs ne doivent jamais se trouver simultanément à l'état « Actif ».

$$VE[IND\_YYY\_Actif] \cap VE[IND\_XXX\_Actif] = \phi \Rightarrow P\_5 = 0 \quad (10)$$
$$VE[IND\_YYY\_Actif] \cap VE[IND\_XXX\_Actif] \neq \phi \Rightarrow P\_5 = 1$$

Dans le second cas :

La propriété de sécurité à vérifier peut s'exprimer par l'incompatibilité conditionnelle entre les indicateurs internes *IND_YYY_Actif* et *IND_XXX_Actif* dans la mesure où un lemme est à vrai. Ainsi, les indicateurs ne doivent jamais se trouver à l'état actif simultanément lorsque le lemme est satisfait.

$$VE[IND\_YYY\_Actif] \cap VE[IND\_XXX\_Actif] \; / \; Lemme = Vrai = \phi \Rightarrow P\_5 = 0, \quad (11)$$
$$VE[IND\_YYY\_Actif] \cap VE[IND\_XXX\_Actif] \; / \; Lemme = Vrai \neq \phi \Rightarrow P\_5 = 1.$$

Les postulats traduisent des séquences avec mémorisation (aspect séquentiel d'une propriété à prouver), ils permettent de positionner une variable lorsque les conditions sont remplies pour qu'une propriété de sécurité soit à prouver. C'est la raison pour laquelle les lemmes font généralement l'objet de graphes particuliers.

Im ersten Fall: Die zu prüfende Sicherheitseigenschaft lässt sich durch die Inkompatibilität zwischen den internen Indikatoren *IND_YYY_Aktiv* und *IND_XXX_Aktiv* ausdrücken. Die zwei Indikatoren dürfen sich nie gleichzeitig im „aktiven" Zustand befinden.

$$VE[IND\_YYY\_Aktiv] \cap VE[IND\_XXX\_Aktiv] = \phi \Rightarrow P\_5 = 0, \quad (10)$$
$$VE[IND\_YYY\_Aktiv] \cap VE[IND\_XXX\_Aktiv] \neq \phi \Rightarrow P\_5 = 1.$$

Im zweiten Fall: Die zu prüfende Sicherheitseigenschaft lässt sich durch die bedingte Inkompatibilität zwischen den internen Indikatoren *IND_YYY_Aktiv* und *IND_XXX_Aktiv* ausdrücken, falls ein Lemma im wahren Zustand ist. Die Indikatoren dürfen nie gleichzeitig im „aktiven" Zustand sein, wenn das Lemma gilt.

$$VE[IND\_YYY\_Aktiv] \cap VE[IND\_XXX\_Aktiv] \; / \; Lemme = Wahr = \phi \Rightarrow P\_5 = 0, \quad (11)$$
$$VE[IND\_YYY\_Actif] \cap VE[IND\_XXX\_Aktiv] \; / \; Lemme = Wahr \neq \phi \Rightarrow P\_5 = 1.$$

Die Anforderungen stellen die Sequenzen mit Erinnerung dar (sequenzieller Aspekt einer zu beweisenden Eigenschaft). Sie erlauben es, den Wert einer Variablen zu stellen, wenn die Bedingungen zum Beweis einer Sicherheitseigenschaft erfüllt sind. Aus diesem Grund werden für die Lemmata spezielle Graphen erstellt.

Il est remarquable de noter que la méthode permet ainsi de prouver simultanément toutes les propriétés au fur et à mesure de l'exploration des états accessibles de l'automate produit du poste d'aiguillage.

### 6.3.4.2 Lien avec les plans de tests avant mise en service des postes

L'automatisation de la preuve repose sur une écriture *a priori*, des propriétés de sécurité à vérifier. Cette opération n'est pas réalisable simplement sur un système quelconque. Cette opération se révèle aisément réalisable pour les postes d'aiguillage. Ceux-ci réalisent les incompatibilités et enclenchements déterminés par la topologie du plan de voie et le programme d'exploitation du poste.

Ces propriétés sont déterminées par des agents spécialisés, itinéraire par itinéraire. Les conditions de sécurité et de fonctionnement doivent être respectées pour que son exploitation soit sûre.

Ainsi, les automates de preuve génériques sont directement instanciés à partir du plan de test définis par les agents spécialisés. Prenons l'exemple des deux itinéraires du cas d'application de la figure 6.23

Es ist bemerkenswert, dass die Methode es somit erlaubt, alle Eigenschaften gleichzeitig zu beweisen, je nach Fortschritt der Auswertung der zugänglichen Zustände des Systemgraphen des Stellwerks.

### 6.3.4.2 Verbindung zu den Testplänen vor Inbetriebnahme der Stellwerke

Die Automatisierung des Beweises beruht auf einer Vorabbeschreibung der zu prüfenden Sicherheitseigenschaften. Dies ist nicht einfach für jedes beliebige System durchzuführen. Für ein Stellwerk ist dies jedoch nicht schwierig, da hierbei die Unvereinbarkeiten und die Sicherungen durch die Topologie des Gleisplans und durch das Betriebsprogramm des Stellwerks vorgegeben sind.

Diese Eigenschaften werden von den Signaltechnikern für jede Fahrstraße bestimmt. Für einen sicheren Fahrbetrieb müssen die Sicherheits- und Funktionsbedingungen eingehalten werden. Daher sind die allgemeinen Beweisautomaten direkt durch den Testplan vorgegeben, der von den Signaltechnikern definiert wird. Als Beispiel sollen die zwei in Abb. 6.23 dargestellten Fahrstraßen dienen.

| **Préparation et Formation de l'itinéraire <00> A→B** | | | | |
|---|---|---|---|---|
| Aiguilles positionnées à Gauche | Aiguilles positionnées à Droite | Itinéraire incompatible de sens opposé | | Effets de la formation de l'itinéraire |
| <01> **AG1** | <02> | <03> **B→A** | | <04> **Tr1I** |
| **Établissement** | | | | |
| Aiguilles contrôlées à Gauche | Aiguilles contrôlées à Droite | Zones de protection | Conditions de la voie de destination | Conditions annexes |
| <05> **AG1** | <06> | <07> **Z1** | <08> | <09> |
| **Destruction Automatique** | | **Destruction manuelle** | | |
| <09> **Z1↓** et <10>**KPd↓** | | <12>**MSG Dest_AB** | | |
| <11> **Z1↑** | | <13> **si Z1↑** | | |

| **Vorbereitung und Fahrstraßenbildung <00> A→C** | | | | |
|---|---|---|---|---|
| Links gestellte Weichen | Rechts gestellte Weichen | Unvereinbarer Streckengegensinn | | Auswirkungen der Fahrstraßen-bildung |
| <01> | <02> **AG1** | <03**> C→A** | | <04> **Tr1I** |
| **Ausführung** | | | | |
| Links kontrollierte Weichen | Rechts kontrollierte Weichen | Deckungs-zonen | Bedingungen des Zielgleises | Zusätzliche Bedingungen |
| <05> | <06> **AG1** | <07> **Z1** | <08> | <09> |
| **Automatische Auflösung** | | **Manuelle Auflösung** | | |
| <09> **Z1↓** et <10>**KPd↓** | | <12>**MSG Dest_AB** | | |
| <11> **Z1↑** | | <13> **falls Z1↑** | | |

**Tableau 6.3** : Propriétés de sécurité formalisées sous la forme d'un cahier d'essais – Cas de l'itinéraire A vers B du plan de voie de la figure 6.23

**Tabelle 6.3**: Sicherheitseigenschaften als Prüfplan formuliert – Fall der Fahrstraße A nach C bezüglich des Gleisplans der Abb. 6.23

La connaissance *a priori* des fonctions d'enclenchement du poste pour chaque phase de fonctionnement permet de définir des automates de preuve formalisant les propriétés de sécurité à respecter pour chacune d'elle. La prise en compte des particularités du poste par le biais du plan d'essais permet de retenir les AP valides et de les instancier avec les valeurs correspondant à l'itinéraire à prouver. Ainsi voici la description d'un AP générique pour la validation des conditions nécessaires pour assurer la sécurité d'une circulation sur l'itinéraire A vers B ou A vers C de la figure 6.33.

Die *a priori* Kenntnis der Sicherungsfunktionen des Stellwerks für jede Funktionsphase erlaubt es, die Beweisautomaten zu definieren, die die einzuhaltenden Sicherheitseigenschaften formalisieren. Die Berücksichtigung der Besonderheiten des Stellwerks mithilfe des Versuchsplans erlaubt es, die gültigen AP beizubehalten und mit den Werten der zu prüfenden Fahrstraße zu parametrisieren. Abb. 6.33 zeigt die Beschreibung eines generischen AP für die Bewertung der notwendigen Bedingungen der Fahrstraße A nach B oder A nach C (Abb. 6.33).

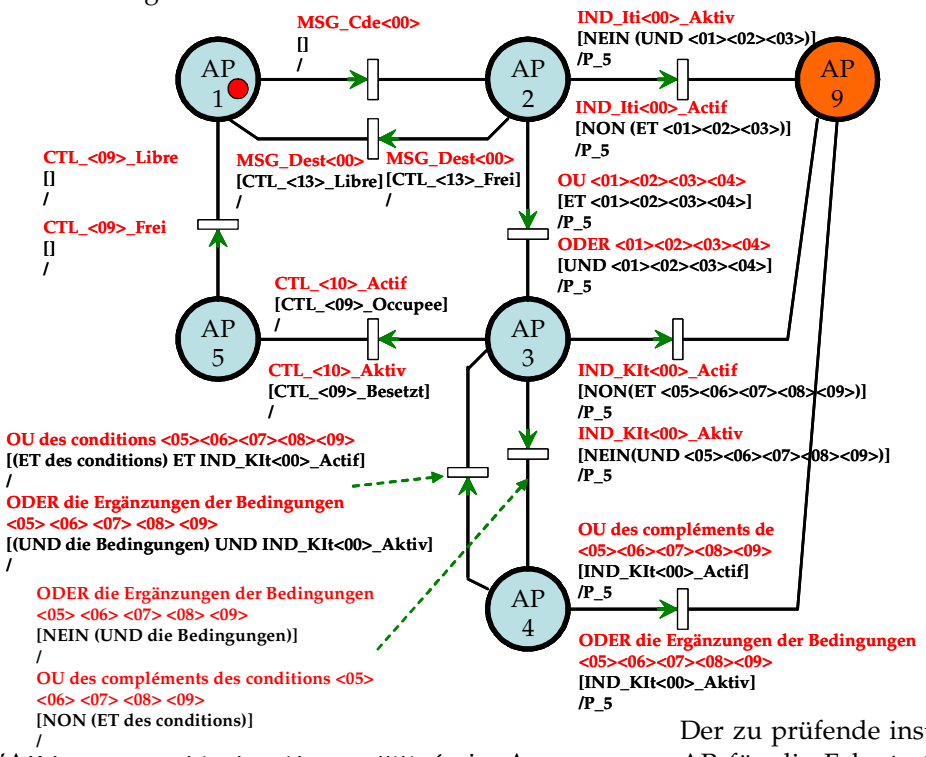



**Figure 6.33** : Automate de preuve générique pour la vérification des propriétés de sécurité d'un itinéraire

**Abbildung 6.33**: Generischer Beweisautomat für die Prüfung der Sicherheiteigenschaften einer Fahrstraße

L'AP à prouver et instancié pour l'itinéraire A vers B devient alors :

Der zu prüfende instanzierte Beweisautomat (AP) AP für die Fahrstraße von A nach B ist dann in Abb.6.34 zu sehen

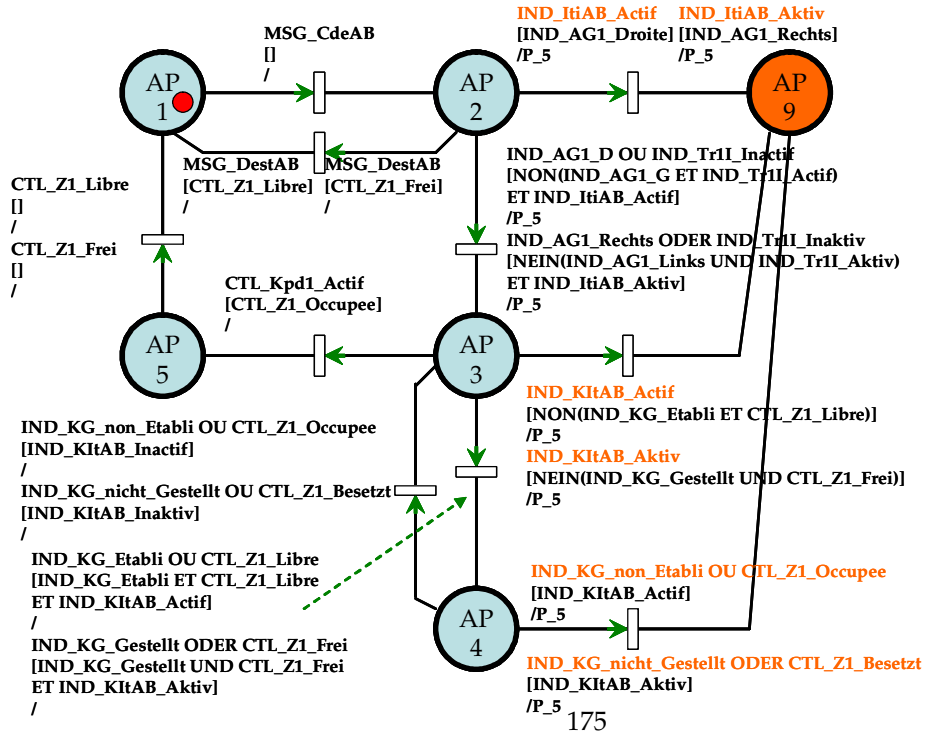



**Figure 6.34** : Automates de preuve instancié pour l'itinéraire A vers B

**Abbildung 6.34**: Für die Strecke von A nach B instanziierte Beweisautomaten

### 6.3.4.3 Postulats et Surabondants

En complément des informations obtenues à partir du plan d'essais, l'analyse de la topologie du plan de voie et de la technologie retenue pour les ressources partagées (circuit de voie, appareil de voie…), les automates de preuve permettent de vérifier des propriétés supplémentaires :

- Les propriétés de surabondance ;
- Les propriétés de non respect des postulats de fonctionnement.

La formalisation de ces propriétés repose sur la formalisation des connaissances «implicites» des agents signalisation.

Donnons des exemples :

Détection d'une surabondance[73] :

- Les conditions nécessaires au contrôle de l'itinéraire sont satisfaites et la commande d'ouverture du signal n'est pas effective ;
- Les conditions nécessaires à la formation d'un itinéraire sont satisfaites et la commande de formation de l'itinéraire n'est pas effective ;
- La séquence nécessaire à la destruction automatique est satisfaite et celle-ci n'a pas lieu…

Non respect des postulats :

- Les circuits de contrôle des aiguilles ne permettent pas (SIL4) que les deux contrôles G et D ne peuvent être établis simultanément ;
- L'action sur le détecteur de passage est utilisée de manière à ne générer qu'une impulsion (montage SIL4 par décharge de capacité)…

### 6.3.4.4 Réalisation pratique de la preuve

La preuve se fait par l'exécution des AP prenant comme entrées les entrées du fonctionnel accompagnées des sorties produites par le fonctionnel suite à l'application de ces entrées. De cette façon, les automates de preuve (AP) permettent de vérifier que le fonctionnel répond correctement aux variations de l'environnement extérieur.

Le moteur de preuve est constitué du moteur de résolution des graphes du fonctionnel auquel sont ajoutées quatre FIFO pour la partie preuve. Les événements externes auxquels sont sensibles les AP sont les entrées terrains (CTL), les messages (MSG) et les indicateurs du fonctionnel (IND) dont le changement de valeur a rendu vrai au moins une transition du fonctionnel et les fins de temporisation (FTP).

[73] La méthode Descubes permet aussi de détecter d'éventuelles conditions d'enclenchement superfues (surabondantes) [Descubes, 1898]

### 6.3.4.3 Anforderungen und Überflüssigkeit

Zusätzlich zu den im Testplan enthaltenen Informationen, der Topologieanalyse des Gleisplans und der ausgewählten Technologie für die geteilten Ressourcen (Gleisstromkreis, Weichen…) erlauben es die Beweisautomaten, die folgenden zusätzlichen Eigenschaften zu überprüfen:

- die Eigenschaften der Überflüssigkeit
- die Eigenschaften der Nichteinhaltung der Funktionsanforderungen.

Die Formalisierung dieser Eigenschaften beruht auf der Formalisierung der „impliziten" Kenntnisse der Signaltechniker. Einige Beispiele:

Entdecken der Überflüssigkeit[74]:

- Die Bedingungen für die Kontrolle der Fahrstraße sind erfüllt und der Befehl für die Öffnung des Signals ist nicht wirksam.
- Die Bedingungen für die Bildung der Fahrstraße sind erfüllt und der Befehl der Fahrstraßenbildung ist nicht wirksam.
- Die notwendige Sequenz für die automatische Auflösung ist erfüllt und die Auflösung findet nicht statt…

Nichteinhaltung der Anforderungen:

- Die Überwachungschaltkreise der Weichen erlauben es nicht (SIL4), dass die zwei Überwachungen L (links) und R (recht) gleichzeitig ausgeführt werden können.
- Das Betätigen des Überfahrtdetektors wird so benutzt, dass nur ein Impuls erzeugt wird (SIL4-Montage durch Kapazitätsentladung)…

### 6.3.4.4 Praktische Umsetzung des Beweises

Der Beweis wird mithilfe der Ausführung von Beweisautomaten durchgeführt, denen man als Eingabe den Eingang der Funktion zusammen mit den Ausgängen gibt, die die Funktion mit eben diesen Eingängen geliefert hat. Auf diese Weise sind die Beweisautomaten (AP) dazu fähig zu prüfen, ob die Funktion bei einer Veränderung der Umwelt richtig reagiert.

Der Beweismotor (die Beweismaschine) besteht aus der Lösungsmaschine der funktionellen Graphen zusammen mit den vier FIFO-Speichern für den Beweis. Die externen Ereignisse, auf die die Beweisautomaten empfindlich reagieren, sind die vor Ort Eingänge (CTL), die Nachrichten (MSG) und die Indikatoren (IND) der Funktion, deren Wertveränderung mindestens eine Transition der Funktionen aktiviert, und das Ende der Verzögerung (FTP).

[74] Die Descubes Methode ermöglich auch die Überflüssige Verschlusse zu detektieren [Descubes, 1898]

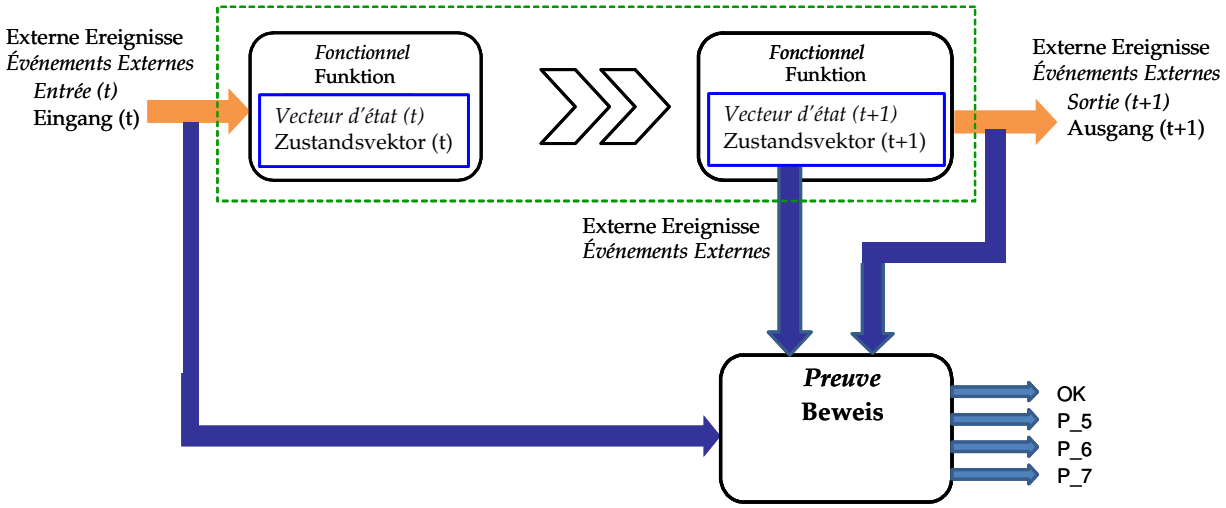



**Figure 6.37** : Principe d'application de la méthode preuve / **Abbildung 6.37**: Anwendungsprinzip der formalen Methode

Le prouveur va ensuite parcourir l'ensemble des états accessibles afin de vérifier qu'aucun d'entre eux ne correspond à une situation contraire à la sécurité. C'est-à-dire que toutes les séquences possibles d'événements externes vont être testées. Il apparaît *a priori* évident que cette méthode est victime de l'explosion combinatoire. Néanmoins, une écriture appropriée des AP permet d'atténuer cet inconvénient et d'obtenir des temps de calcul acceptables par rapport aux contraintes temporelles imposées lors d'un processus d'essai, pour la mise en service d'un PIPC par exemple.

Voir l'annexe D pour une description plus précise.

### 6.3.4.5 Les automates de preuve (AP)

Les automates de preuve constituent la partie la plus délicate de la méthode car d'eux vont dépendre de :

- la validité de la preuve : est-ce que ce qui est prouvé est nécessaire et suffisant pour que la sécurité soit garantie ?
- le temps de calcul nécessaire pour effectuer la preuve ;
- l'adoption de la méthode. La généricité des AP doit être assurée pour que la méthode soit exploitable par les experts en signalisation.

Au franchissement d'une transition d'un AP permet de mettre à jour des indicateurs spécifiques: P_5, P_6 ou P_7 (Figure 6.38).

- P_5 traduit le fait qu'une propriété de sécurité n'est pas remplie ;
- P_6 traduit le fait qu'un des postulats n'est pas respecté ;
- P_7 traduit le fait qu'il existe une condition d'enclenchement surabondante.

Der Beweismotor durchläuft dann alle zugänglichen Zustände, um zu prüfen, ob einer davon einer unsicheren Situation entspricht. Das bedeutet, dass alle möglichen externen Ereignisse getestet werden. Es scheint von vornherein klar zu sein, dass diese Methode zu einer kombinatorischen Explosion führt. Trotzdem ermöglicht es eine geeignete Formulierung der Beweisautomaten, diesen Effekt abzuschwächen und für den zeitlichen Rahmen des Versuchsablaufs der Inbetriebnahme eines PIPC-Stellwerks annehmbare Rechenzeiten zu erhalten.

Siehe Anhang D für mehr präzise Beschreibung.

### 6.3.4.5 Beweisautomaten (AP)

Die Beweisautomaten stellen den kritischsten Teil der Methode dar, denn von ihnen hängt ab:

- die Gültigkeit des Beweises: sind die durchgeführten Beweise notwendig und hinreichend, um die Sicherheit zu garantieren?
- die Rechenzeit für den Beweis.
- die Akzeptanz der Methode. Die Erzeugbarkeit der AP muss gewährleistet werden, damit die Methode von den Signaltechnikexperten nutzbar ist.

Die Schaltung einer Transition eines AP erlaubt es, die spezifischen Indikatoren P_5, P_6 oder P_7 zu aktualisieren (Abb. 6.38):

- P_5 zeigt an, dass eine Sicherheitseigenschaft nicht erfüllt ist.
- P_6 zeigt an, dass eine der Anforderungen nicht respektiert wird.
- P_7 zeigt an, dass es eine überflüssige Sicherungsbedingung gibt, die unter normalen Bedingungen zu keiner unsicheren Situation führt, aber bei Missachten von Vorschriften zu einer Gefahr führen kann.

Le changement d'état d'un de ces indicateurs ne peut être utilisé comme événement par les autres AP. Ils sont donc utilisés pour contrôler qu'une séquence attendue s'est ou non correctement déroulée. Cette contrainte traduit le fait que les AP sont des observateurs passifs, sans interactions et sans action sur les graphes fonctionnels.

Cette contrainte a aussi pour effet d'éviter que les automates de preuves deviennent le miroir des automates fonctionnels. Il s'agit en effet de se focaliser uniquement sur ce que doit réaliser le fonctionnel et non la manière dont il le fait.

Les contraires de la sécurité (P_5) : La détection des situations contraires à la sécurité est la raison d'être de la méthode de preuve. Lorsqu'une telle situation se produit, l'AP positionne l'indicateur P_5 à 1. Une façon de raisonner est d'effectuer une action P_5 dès qu'une incompatibilité est observée.

Les conditions surabondantes (P_7) : Il s'agit de vérifier qu'il n'existe pas d'enclenchement superflu, celui-ci pouvant conduire à une situation dangereuse. Par exemple, un itinéraire dont le signal de protection resterait toujours fermé alors que l'itinéraire est formé et contrôlé n'est pas contraire à la sécurité. En effet, aucun train ne pourrait emprunter l'itinéraire. Il faut pourtant signaler que les graphes fonctionnels ne répondent pas aux règles de la signalisation qui veulent que le signal s'ouvre alors.

Les postulats (P_6) : L'exploration de tous les états accessibles donne lieu à une explosion combinatoire qui pourrait rendre rédhibitoire l'utilisation de la méthode sur des systèmes industriels. Les postes d'aiguillage sont régis par une règlementation précise et des équipements terrains de sécurité. Il est alors possible d'en tirer parti afin d'échapper à cette l'explosion combinatoire.

La figure 6.38 récapitule le positionnement des indicateurs précédents dans l'ensemble des états accessibles en relation avec le processus de preuve. L'écriture des automates de preuve requiert des connaissances métier que nous avons longuement décrit dans le chapitre 4. Il s'agit de décrire sous forme d'automate :

- les configurations et risques et événements redoutés du système ferroviaire (Tableau 4.2) : indicateurs P_5
- les conditions d'exploitation du système ferroviaire, tant en mode nominal qu'en mode dégradé : indicateurs P_6
- les fonctionnalités des postes d'aiguillage nécessaires à l'exploitation du système ferroviaire : indicateurs P_7

Die Zustandsänderung eines dieser Indikatoren kann nicht als Ereignis für einen anderen AP benützt werden. Sie werden nur benützt, um zu kontrollieren, ob eine erwartete Sequenz richtig ausgeführt wurde oder nicht. Diese Beschränkung zeigt, dass die AP passive Beobachter sind, ohne Wechselwirkungen und ohne Handlungsmöglichkeit in Bezug auf die funktionellen Graphen. Diese Beschränkung vermeidet auch, dass die AP zu Spiegeln der funktionellen Graphen werden. Man konzentriert sich nur darauf, was von der Funktion durchgeführt werden soll und nicht wie es durchgeführt wird.

Die Sicherheitsrandbedingungen (P_5): das Erkennen der im Widerspruch zur Sicherheit stehenden Situationen ist die Daseinsberechtigung der Beweismethode. Wenn eine solche Situation eintritt, stellt der AP den Indikator P_5 auf 1 um. Eine Möglichkeit für den Beweis ist die, die Aktion P_5 sofort auszuführen sobald eine Unverträglichkeit entdeckt wurde.

Das überflüssige Ereignis (P_7): Es genügt nicht nur zu prüfen, dass keine unsichere Situation eintreten kann, da eine gefährliche Situation von einer überflüssigen maskiert werden kann. Zum Beispiel steht ein Hauptsignal, das immer auf rot steht, selbst wenn die Fahrstraße gebildet und kontrolliert ist, nicht im Widerspruch zur Sicherheit. In der Tat kann so kein Zug diese Fahrstraße benützen. Es ist hierbei zu unterstreichen, dass die funktionellen Graphen nicht den Regeln der Signalgebung folgen, nach denen das Signal geöffnet wird.

Die Anforderung (P_6): Die Auswertung aller zugänglichen Zustände führt zu einer kombinatorischen Explosion, die die Benutzung der Methode auf Industriesystemen unmöglich machen könnte. Die Stellwerke unterliegen einer strengen Reglementierung und sind vor Ort mit Sicherheitsanlagen ausgerüstet. Es ist möglich, diese Tatsache auszunützen und so der kombinatorischen Explosion zu entgehen.

Abb. 6.38 fasst die verschiedenen möglichen Werte der zuvor aufgeführten Indikatoren, die mit dem Beweisvorgang in Verbindung stehen, zusammen. Für die Beweweisautomat Beschreibung braucht man eine feste Fachkenntnis. Es handelt sich darum in Form von Automaten:

- die Konfiguration, die gefürchteten Risiken und die Ereignisse des Eisenbahnsystems (Tabelle 4.2): Indikatoren P_5,
- die Ausnutzungsbedingungen (Betriebsbedingungen) von Eisenbahnsystems, sowohl in der normalen als auch in der degradierten Weise: Indikatoren P_6,
- die notwendigen Betriebsfunktionalitäten der Stellwerke: Indikatoren P_7,

zu beschreiben.

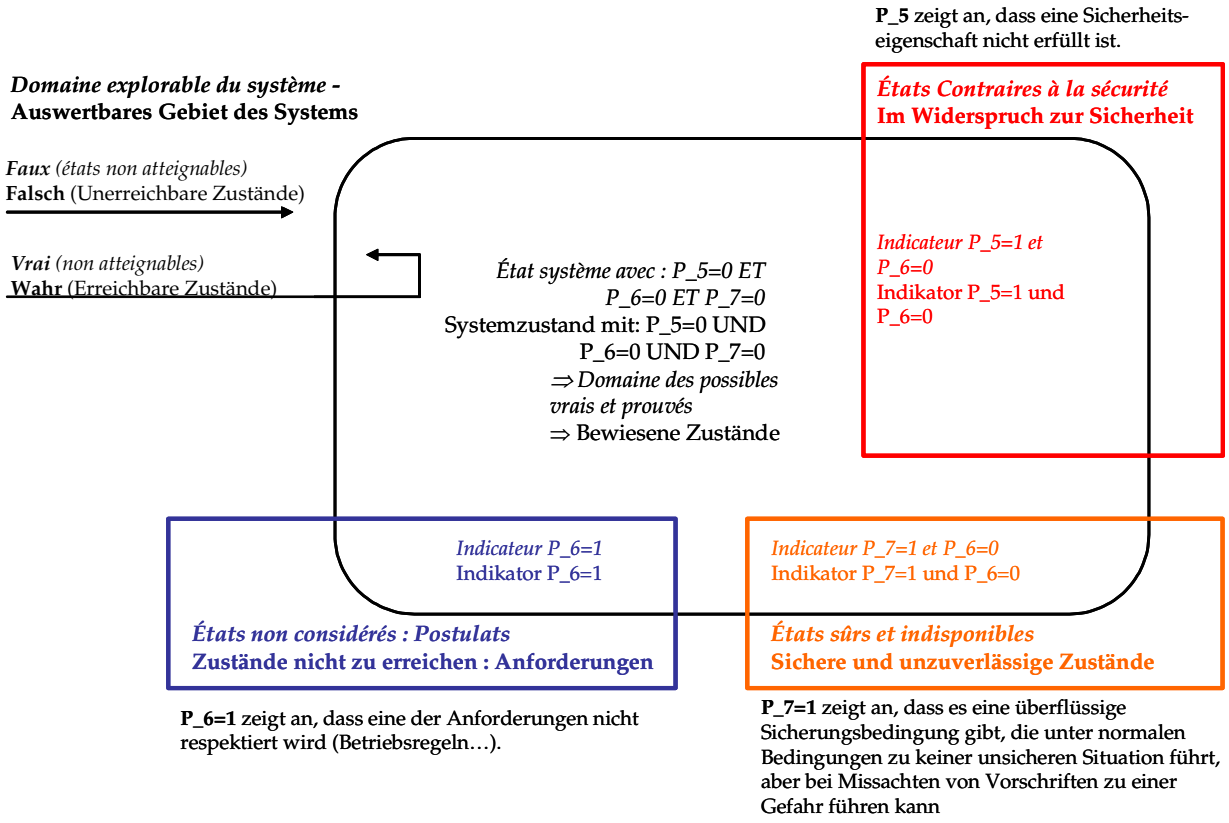



**Figure 6.38** : Positionnement des indicateurs dans l'ensemble des états du système
**Abbildung 6.38**: Werte der Indikatoren in allen Zuständen des Systems

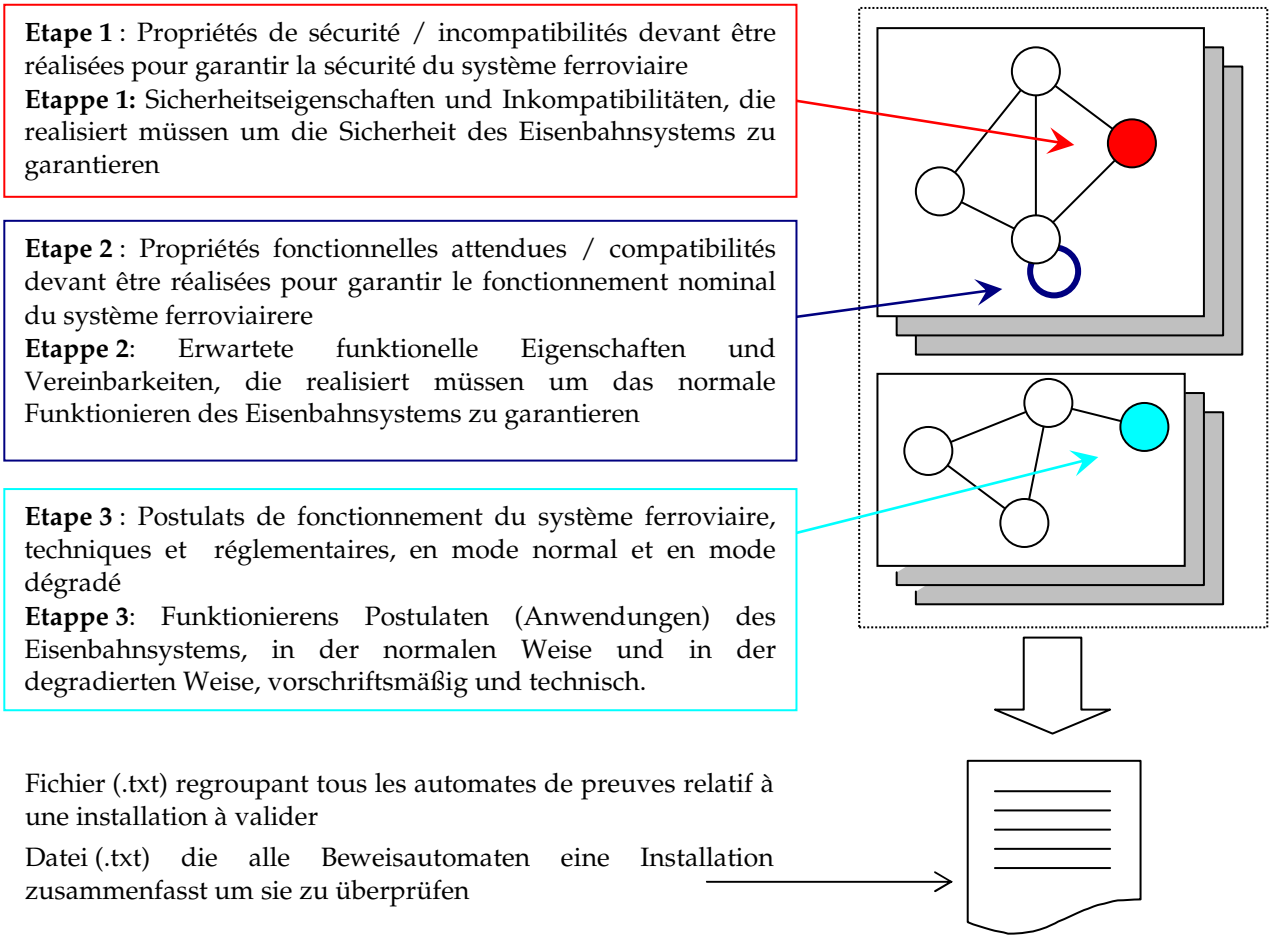



**Figure 6.39** : Démarche d'écriture des automates de preuve
**Abbildung 6.39**: Schreibung den Beweisautomaten Procedure

L'écriture des automates de preuve suit la démarche générale suivante (exemple figure 6.32) :

- Etape 1 : Ecrire sous forme d'automates les séquences d'événements externes, qui traduisent l'apparition d'une situation redoutée (indicateurs P_5) ;
- Etape 2 : Compléter les automates précédents par l'écriture des séquences qui traduisent l'absence de fonctionnalités attendues du système (indicateurs P_7);
- Etape 3 : Ecriture sous forme de nouveaux automates des séquences d'événements externes qui ne sont pas autorisées par la réglementation ou pas possibles du fait de l'environnement technique du système informatique (indicateurs P_6).

La figure 6.39 illustre la démarche :

## 6.4 Ce qu'il faut retenir pour la suite du travail

L'état de l'art montre qu'il est très difficile, voire impossible, d'intégrer des modèles détaillés des systèmes embarqués dans un modèle probabiliste d'installation. La stratégie de modélisation des automatismes proposée consiste alors à opter pour des représentations probabilistes partielles et contextualisées des systèmes embarqués en deux couches :

- une couche matérielle et logiciel d'exploitation (entrées, sorties, sécurité, communication) indépendant des fonctionnalités portées ;
- une couche logicielle fonctionnelle sous forme d'automate à états finis, qui seront intégrés dans un modèle plus général (tout ou partie d'un système de transport, d'une installation industrielle..).

**A cette fin nous avons proposé une méthode de validation formelle de la correction des fonctionnalités portées par l'ensemble afin de garantir que seuls les défauts probabilistes de la première couche sont à prendre en compte dans les calculs requis par les normes.**

**L'application en temps réel par la machine cible des spécifications prouvées permet d'étendre la preuve à l'ensemble du cycle de développement.**

Die Schreibung der Beweisautomaten folgt die allgemeinen Procedure (Beispiel Figur 6.32):

- Etappe 1: In Form von Automaten die äußerlicher Ereignisse Sequenzen schreiben, die die das Erscheinen einer gefürchtete Situation übersetzten (Indikatoren P_5),
- Etappe 2: Die vorigen Automaten von der Schrift der äußerlicher Eingänge Sequenzen ergänzen, die die Abwesenheit vom System erwarteter Kunktionalitäten übersetzen (Indikatoren P_7),
- Etappe 3: In Form von neuen Automaten die äußerlicher Ereignisse Sequenzen schreiben, die entweder, von der Reglementation nicht gestattet werden, oder wegen der Umwelt des Systems nicht realistisch sind (Indikatoren P_6).

Die Figur 6.39 illustriert die Procedure:

## 6.4 Zusammenfassung

Der Kenntnisstand zeigt, dass es sehr schwierig wenn nicht gar unmöglich ist, detaillierte Modelle von on-board Systemen in ein stochastisches Modell der Anlage zu integrieren. Die vorgeschlagene Strategie der Steuerungsmodellierung besteht deshalb darin, partielle stochastische und kontextuelle Darstellungen bei on-board Systemen zu bevorzugen. Dies geschieht auf zwei Ebenen:

- eine materielle Ebene und eine Betriebssoftwareebene (Eingänge, Ausgänge, Sicherheit, Kommunikation) unabhängig von den vorhandenen Funktionen,
- eine Funktionssoftwareebene in Form von endlichen Automaten, die später in ein allgemeines Modell integriert werden (ein ganzes Transportsystem oder ein Teil davon, eine Industrieanlage…).

**Hierfür wurde eine Methode der formalen Validierung vorgestellt, die das ganze System betreffende Funktionen korrigiert, um sicherzustellen, dass bei den von den Normen geforderten Berechnungen nur noch die stochastischen Fehler der ersten Ebene berücksichtigt werden müssen. Durch die Echtzeitanwendung der bewiesenen Spezifizierungen auf der Zielmaschine kann der Beweis auf den ganzen Entwicklungszyklus ausgedehnt werden.**

Les hypothèses suivantes sont indispensables afin d'élaborer la preuve :

- Le fonctionnement de l'interpréteur est cyclique et déterministe d'un point de vue fonctionnel et temporel ;
- Le fonctionnement de l'interpréteur du fonctionnel, c'est à dire l'algorithme de résolution et de stabilisation du réseau des automates est événementiel (sans perte d'événement quelque soit la séquence des événements et l'algorithme de résolution du marquage) ;
- Tout changement d'état de l'environnement se traduit par un événement externe (changement d'état d'un équipement terrain, message, échéance d'une temporisation...) ;
- Il ne peut se produire qu'un seul et unique événement à la fois. Le nombre d'événements est fini dans un intervalle de temps borné. En effet, si plusieurs événements pouvaient se produire simultanément, on ne connaîtrait pas la façon dont cette situation serait traitée par les automates ;
- Le temps n'existe pas : d'une part le temps est décrit par le numéro d'ordre des instants correspondant aux événements ; d'autre part seule est retenue la notion de temporisation à trois états : non armée, armée en cours, échue.
- Il existe un unique événement à la fois traité par l'interpréteur. Lorsque qu'un événement est injecté dans le réseau d'automates : les transitions, le marquage, les conditions de franchissement et les actions associées sont traités jusqu'à stabilisation de l'évolution du réseau d'automates. Puis l'événement suivant est traité. Ainsi toutes les actions relatives à chaque événement sont traitées avant la prise en compte de nouveaux événements;
- A chaque événement, l'ordre de prise en compte des automates est fixe pour une application donnée. Il est nécessaire de connaître cet ordre, et le même sera pris en compte pour l'élaboration de la liste des états accessibles du système (pour la preuve formelle). Cet ordre ne doit pas influer sur le comportement fonctionnel du système.

Die folgenden Hypothesen sind für die Ausarbeitung des Beweises unerlässlich:

- Das Funktionieren des Interpreters ist vom funktionellen und zeitlichen Standpunkt aus zyklisch und deterministisch. Das Funktionieren des Interpreters der Funktionen, d.h. der Lösungsalgorithmus und der Algorithmus für die Stabilisierung des Automatennetzes, sind ereignisbasiert (d. h. es wird kein Ereignis verpasst, wie immer die Ereignissequenz und der Lösungsalgorithmus der Markierung auch aussehen). Jede Änderung des Umgebungszustandes wird in Form eines externen Ereignisses (Änderung des Zustandes einer Anlage vor Ort, Nachricht, Ende einer Verzögerung) ausgedrückt. Es kann nur ein einziges Ereignis zugleich auftreten. Die Anzahl der Ereignisse ist in einer abgegrenzten Zeitspanne endlich. Wenn mehrere Ereignisse gleichzeitig auftreten könnten, wüsste man nicht, wie diese Situation von den Automaten behandelt werden würde,
- Zeit existiert nicht: Einerseits wird die Zeit durch die Ordnungszahl der den Ereignissen entsprechenden Zeitpunkten beschrieben; andererseits wird nur der Begriff der Verzögerung in drei Ausprägungen benutzt: nicht gestellt - gestellt und aktiv - beendet. Vom Interpreter wird nur ein einziges Ereignis auf einmal bearbeitet.
- Wenn ein Ereignis ins Automatennetz eingelesen wird: Die Transitionen, die Markierung, die Schaltungsbedingungen und die assoziierten Handlungen werden bis zur Stabilisierung der Veränderung des Automatennetzes bearbeitet. Danach wird das folgende Ereignis behandelt. Auf diese Weise werden alle zu dem Ereignis gehörenden Handlungen bearbeitet, bevor ein neues Ereignis berücksichtigt wird,
- Die Reihenfolge, in der die Automaten ein Ereignis in einer gegebenen Anwendung berücksichtigen, ist festgelegt. Es ist notwendig, diese Reihenfolge zu kennen. Dieselbe Reihenfolge wird für die Ausarbeitung der Liste der erreichbaren Systemzustände (für den formalen Beweis) benützt. Diese Reihenfolge darf das funktionelle Verhalten des Systems nicht beeinflussen.

# CHAPITRE 7

## Applications de notre méthode à des situations de postes réels

## 7.1 Introduction

Illustrons l'application des enseignements des chapitres précédents à des cas industriels ferroviaires réels. A cette fin nous traiterons les situations suivantes :

- Les postes mécaniques : Il s'agit de valider formellement les fonctions de contrainte réalisées par ces postes. La méthode sera appliquée à une modélisation de ces installations anciennes. Il apparaît que des méthodes formelles étaient déjà appliquées dans le passé sous un autre formalisme [Plisson, 1886] [Descubes, 1898] ;
- Les postes informatiques de type PIPC[75] : il s'agit de valider formellement les automatismes réalisés par des postes réels en exploitation en France. Notre méthode sera appliquée à un fonctionnel réel, interprété en temps réel par la structure d'accueil commune des PIPC (modules d'enclenchement) ;
- Les installations futures : Elles reposent sur des automates de sécurité. Nous appliquerons la démarche à un passage à niveau de double voie.

D'autres situations ont été traitées, leur nombre et les contraintes de confidentialité qui y sont attachées interdisent d'en réaliser un inventaire exhaustif.

[75] Le fonctionnement de ce type de poste d'aiguillage est décrit précisément dans [Antoni, 2009-1] et [Antoni, 2009-4]

# KAPITEL 7

## Anwendung der neuen Methode auf echte Stellwerke

## 7.1 Einleitung

In diesem Kapitel wird die Umsetzung der in den vorherigen Kapiteln erlangten Kenntnisse auf wirkliche Beispiele der Eisenbahnindustrie illustriert:

- Mechanische Stellwerke: Es handelt sich darum, die von diesen Stellwerken verwirklichten Beschränkungen formal zu prüfen. Die in dieser Arbeit vorgestellte Methode wird auf ein Modell dieser alten Anlagen angewendet. Es kommt zum Vorschein, dass in der Vergangenheit schon formale Methoden benutzt wurden, jedoch mit einer anderen Formalisierung [Plisson, 1886] [Descubes, 1898].
- Die Rechnerstellwerke vom Typ PIPC[76]: Es handelt sich darum, die von wirklichen, in Frankreich funktionierenden Stellwerken durchgeführten Automatismen formal zu prüfen. Die in dieser Arbeit vorgestellte Methode wird auf eine reelle Funktion angewandt, die von der gemeinsamen Eingangsstruktur des PIPC (MEI-Sicherungsmodul) in Echtzeit interpretiert wird.
- Zukünftige Anlagen: Sie beruhen auf Sicherheitsautomaten. Die Methode wird auf einen zweigleisigen Bahnübergang angewandt.

Andere Fälle wurden mit der neu vorgestellten Methode bearbeitet, ihre große Anzahl und die Vertraulichkeit erlaubten es jedoch nicht, diese hier alle aufzuführen.

[76] Eine präzis Beschreibung diese elektronische Stellwerke steht in [Antoni, 2009-1] und [Antoni, 2009-4]

## 7.2 Poste mécanique

### 7.2.1 Principes de base – Notations Cossmann Descubes

Les postes mécaniques ne constituent pas de véritables automatismes au sens où l'on peut l'entendre aujourd'hui. Ils réalisent en fait des fonctions de contrainte entre les différents organes de commande des ressources gérées dans une zone d'action donnée. En réponse à loi de Murphy [«*s'il est possible que quelque chose se passe mal, alors cela arrivera*»] ces fonctions de contrainte réalisent les incompatibilités nécessaires entre organes de commande (les leviers en l'occurrence) afin que ces combinaisons dangereuses des positions de ces organes ne puissent jamais se produire, quelle qu'en soit la probabilité d'apparition [*une fausse manœuvre est considérée comme sûre si elle est possible*].

La sécurité telle qu'elle a été conçue repose sur la fiabilité de quelques lois physiques et sur une conception «orientée» de la table d'enclenchement. Ainsi, une pièce métallique ne peut pas se déformer sous la contrainte.

Les fonctions de contrainte sont décrites sous la forme d'incompatibilités [Descubes, 1898] [SNCF, 1963] : traduction des positions interdite (à interdire) de deux (ou plusieurs) organes de commande du poste. Il est à noter que les incompatibilités décrivent des combinaisons de positions d'organes et non des arrangements (séquences) de positions d'organes.

Le langage communément utilisé en France depuis le début du 20ème siècle est celui de «Cossmann Descubes». Ce mode d'écriture utilise une propriété générique de ces postes : «*parmi un groupe de leviers non totalement indépendant, un unique levier peut être en mouvement à un moment donné*». [Descubes, 1898] [SNCF, 1963]

Montrons que la table mécanique d'enclenchement ainsi obtenue est équivalente à un automate concurrentiel à contrainte et peut donc se voir appliquée notre méthode de validation formelle. Ce qui a été fait, sous une autre forme, et continue de se faire sur tous les postes mécaniques actuellement en exploitation en France lors d'une évolution de leurs fonctions.

## 7.2 Mechanisches Stellwerk

### 7.2.1 Grundsätze - Cossmann Descubes Schreibweise

Die mechanischen Stellwerke stellen keine Regelungen im heutigen Sinne dar. Sie setzen Beschränkungen zwischen den verschiedenen Befehlsorganen der Ressourcen in einem bestimmten Handlungsraum um. Als Antwort auf Murphys Gesetz [„*wenn es möglich ist, dass etwas schief gehen kann, geht es auch schief*"] verwirklichen diese Beschränkungen die notwendigen Unvereinbarkeiten zwischen den Befehlsorganen (in diesem Fall zwischen den Stellhebeln). Dies wird eingesetzt, um jede mögliche gefährliche Kombination der Stellungen dieser Befehlsorgane zu vermeiden, unabhängig von der Auftrittswahrscheinlichkeit *[ein falsches Manöver wird als sicher betrachtet, wenn es möglich ist].*

Die so entwickelte Sicherheit beruht auf der Zuverlässigkeit einiger physikalischer Gesetze und auf dem „orientierten" Entwurf der Verschlusstische. Ein Stück Metall kann sich nicht ohne Zwangseinwirkung verformen.

Die Beschränkungen werden in Form von Unvereinbarkeiten [Descubes, 1898] [SNCF, 1963] beschrieben: Umsetzung der verbotenen (zu verbietenden) Stellungen von zwei (oder mehreren) Befehlsorganen des Stellwerks. Es ist wichtig anzumerken, dass die Unvereinbarkeiten Kombinationen von Befehlsorganstellungen beschreiben und nicht die Stellung (Sequenz) der Organe.

Die in Frankreich seit Anfang des 20. Jahrhundert allgemein benutzte Sprache ist die von „Cossmann Descubes". Diese Sprache benützt eine allgemeine Eigenschaft dieser Stellwerke: "*Von einer nicht vollständig unabhängigen Gruppe von Stellhebeln kann zu einem Zeitpunkt immer nur einer in Bewegung sein*". [Descubes, 1898] [SNCF, 1963]

Es wird gezeigt, dass ein so erhaltener mechanischer Verschlusstisch zu einem Automaten mit konkurrierenden Zuständen und Randbedingungen äquivalent ist und so die hier vorgestellte Methode für die formale Validierung angewendet werden kann. Dies wurde schon früher unter einer anderen Form gemacht und wird auch heute noch bei allen in Betrieb stehenden mechanischen Stellwerken durchgeführt, sobald diese eine Funktionsänderung erfahren.

Chaque levier peut avoir trois positions distinctes :

- position droite ou normale (N ou +) : la ressource commandée par le levier est alors en position normale (signal fermé, aiguille en position voie directe…), restrictive et/ou sûre et immobilisée dans cette position ;
- position pendant la course (PC ou ±) : la ressource commandée est ou peut être en mouvement, ce durant un temps indéfini ;
- position renversée (R ou -) : la ressource commandée par le levier est alors en position d'ouverture et/ou renversée (signal ouvert, aiguille en position voie déviée…) et immobilisée dans cette position.

### 7.2.1.1 Deux leviers L1 et L2

Prenons deux leviers L1 et L2 d'un même poste (dépendant d'une même table d'enclenchement) :

- si les deux leviers sont reliés par une ou plusieurs relations d'incompatibilités (directe ou résultante à travers d'autres leviers), seul un des leviers peut être en mouvement à la fois ;
- l'état initial du poste est toujours défini comme étant «*tous les leviers peuvent être simultanément en position normale*», dans notre cas L1N et L2N.

Réalisons une incompatibilité qui interdise le renversement simultané des deux leviers. **Cette incompatibilité (L1- L2-) se traduit alors par deux enclenchements** qu'il conviendra de réaliser au moyen de la table d'enclenchement :

$$\textbf{(L1- L2-)} \Leftrightarrow \textbf{\{L1R} \Rightarrow \textbf{L2N et L2R} \Rightarrow \textbf{L1N\}} \qquad \textbf{(13)}$$

Ceci se traduit dans le cadre d'une poste MU45 par la réalisation suivante (figure 7.1) :

Jeder Hebel kann drei verschiedene Positionen haben:

- Gerade oder normale Position (N oder +): die durch den Hebel betätigte Ressource ist in normaler Position (geschlossenes Signal, Weiche in direkter Position…), restriktiven und/oder sicheren Position und in dieser Stellung blockiert.
- Position während der Betätigung (PC oder ±): die betätigte Ressource ist in Bewegung oder kann in Bewegung sein und das für unbestimmte Zeit.
- Umgestellte Position (R oder -): die durch den Hebel betätigte Ressource befindet sich in der offenen oder umgestellten Position (offenes Signal, Weiche in der abzweigenden Position,…) und ist in dieser Stellung blockiert.

### 7.2.1.1 Zwei Hebel L1 und L2

Seien L1 und L2 zwei Hebel desselben Stellwerks (desselben Stelltisches):

- Wenn die zwei Hebel durch ein oder mehrere Unvereinbarkeitsbeziehungen verbunden sind (direkt oder durch andere Hebel), dann kann nur ein einziger Hebel in Bewegung sein.
- Der Anfangsstand des Stellwerks ist immer als der Zustand definiert, in dem alle Hebel in der normalen Position sind (in dem vorliegenden Fall L1N und L2N).

Eine Unvereinbarkeit, die das gleichzeitige Umstellen der zwei Hebel verbietet **(L1- L2-), wird durch zwei Verschlüsse umgesetzt** die man auf einem Stelltisch unterbringen kann:

$$\textbf{(L1- L2-)} \Leftrightarrow \textbf{\{L1R} \Rightarrow \textbf{L2N und L2R} \Rightarrow \textbf{L1N\}} \qquad \textbf{(13)}$$

Für das Stellwerk MU45 sieht die Umsetzung wie in Abb. 7.1 aus.

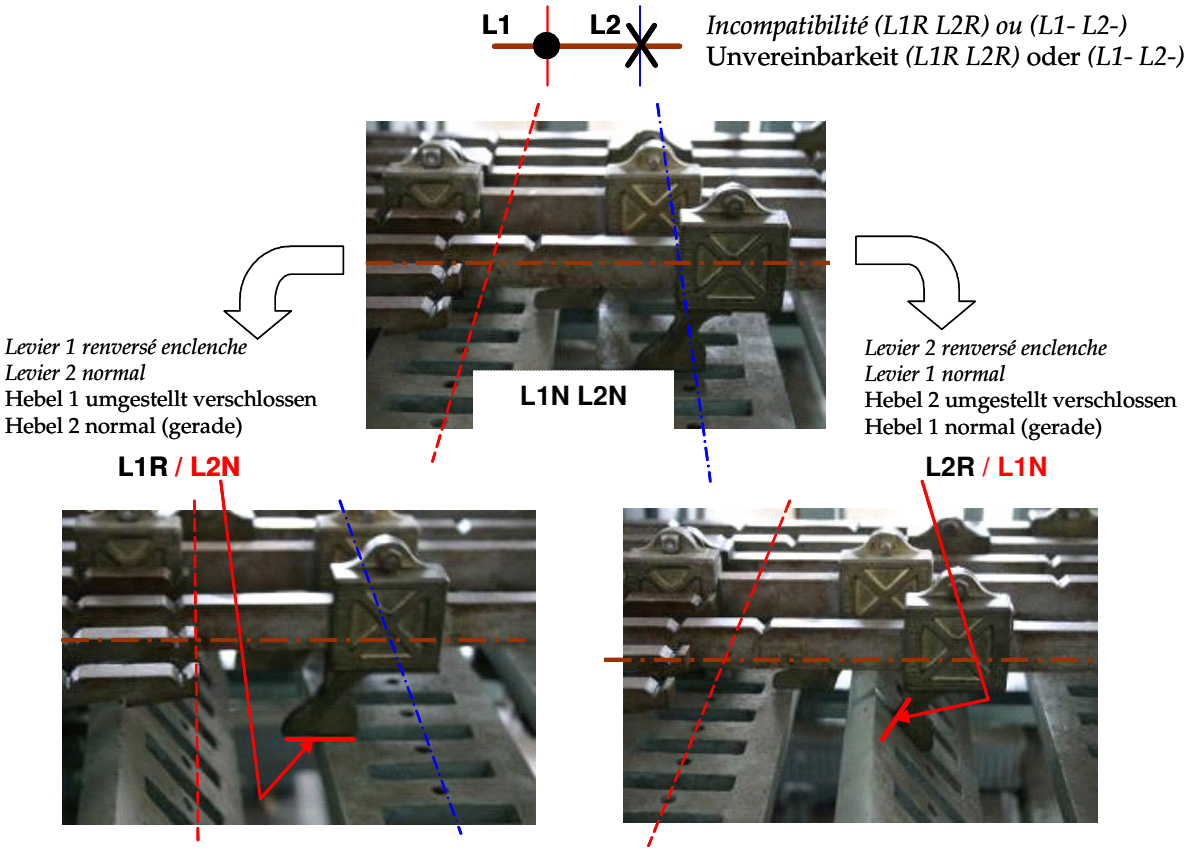
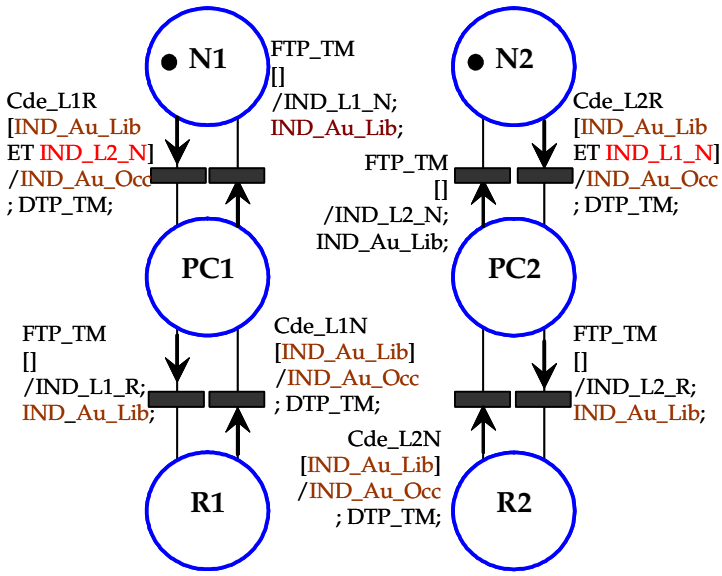



**Figure 7.1 :** Les trois positions du taquet et des grils réalisant l'incompatibilité (L1- L2-)
**Abbildung 7.1:** Die drei Positionen des Verschlussstücks und der Register, die die Unvereinbarkeit (L1- L2-) umsetzen

Ceci peut se traduire de la manière suivante :

Diese Umsetzung kann wiederum wie in Abb. 7.2 schematisiert werden kann.




**Lib = Libre = Frei**
**Occ = Occupé = Besetzt**

**ET=UND**
**OU = ODER**

La notation des réseaux de Petri est celle de E.Grötsch §3.3 dans [Schnieder, 1992]

***Levier 1 : L1***
**Hebel 1: L1**

***Levier 2 : L2***
**Hebel 2: L2**

Die Notation der Netze von Petri ist diejenige von E.Grötsch §3.3 in [Schnieder, 1992]

**Figure 7.2 :** Modélisation des positions de deux leviers L1 et L2 à l'aide des réseaux de PETRI
**Abbildung 7.2**: Modellierung der Positionen der zwei Hebel L1 und L2 durch Petrinetze

Le réseau de la figure 7.2 illustre une première transformation utilisant les notations décrites au Chapitre 6 et les trois indicateurs suivants (chacun ne pouvant prendre que deux positions) :

- **IND_Au** en état Lib (libre) ou Occ (occupé) ;
- **IND_L1** en état N (normal) ou R (renversé) ;
- **IND_L2** en état N (normal) ou R (renversé).

Si l'on élimine le temps de manœuvre (variable aléatoire TM décrivant le temps de renversement d'un levier) et si l'on considère que les graphes sont interprétés selon les règles du langage AEFD (Chapitre 6) nous obtenons le modèle final de la figure 7.3.

Cette transformation montre que seul 4 états fonctionnels significatifs sont à considérer pour décrire le fonctionnel des deux leviers L1 et L2.

L'exploration des états système des graphes initiaux et finaux va produire des nombres sensiblement différents d'états système. Ceci alors que seuls trois états fonctionnels existent réellement ([L1+ L2+], [L1+ L2-], [L1- L2+]) faisant apparaître l'incompatibilité réalisée à savoir (L1- L2-).

Das Netz der Abb. 7.2 auf der rechten Seite zeigt eine erste Umwandlung entsprechend der in Kapitel 6 eingeführten Schreibweise und den drei folgenden Indikatoren (jeder Indikator kann nur zwei Stellungen einnehmen):

- **IND_Au** im freien (Lib) oder belegten (Occ) Zustand
- **IND_L1** im normal (N) oder umgestellten (R) Zustand
- **IND_L2** im normal (N) oder umgestellten (R) Zustand.

Wenn man die Bedienzeit ignoriert (Zufallsvariable TM, die die Zeit der Umstellung eines Hebels beschreibt) und wenn man annimmt, dass die Graphen nach den AEFD-Regeln interpretiert werden (Kapitel 6), erhält man das Endmodell aus Abb. 7.3.

Diese Umwandlung zeigt, dass nur vier wichtige funktionelle Zustände nötig sind, um die Funktionen der zwei Hebel L1 und L2 zu beschreiben. Die Auswertung der Systemzustände der anfänglichen und der endgültigen Graphen ergibt eine gänzlich andere Anzahl von erzeugten Systemzuständen und dies obwohl nur drei funktionelle Zustände wirklich existieren ([L1+ L2+], [L1+ L2-], [L1- L2+]). Die verwirklichte Unvereinbarkeit nämlich (L1- L2-) darstellt.

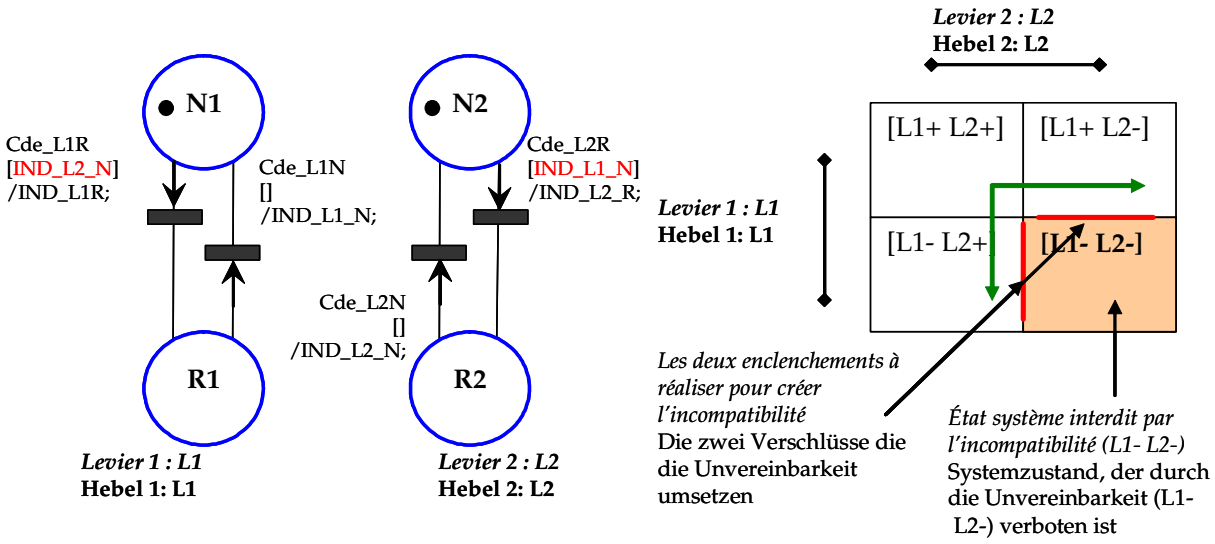



**Figure 7.3** : Modèle final pour une incompatibilité (L1- L2-)
**Abbildung 7.3**: Endmodell für eine Unvereinbarkeit (L1- L2-)

### 7.2.1.2 Trois leviers L1, L2 et L3

Le même raisonnement appliqué à l'incompatibilité (L1+ L3-) en complément de celle (L1- L2-) conduirait au graphe de la figure 7.4.

### 7.2.1.2 Drei Hebel L1, L2 und L3

Dieselbe Überlegung zusätzlich zu der Unvereinbarkeit (L1- L2-) auf die Unvereinbarkeit (L1+ L3-) angewendet, führt zum Graph der Abbildung 7.4.

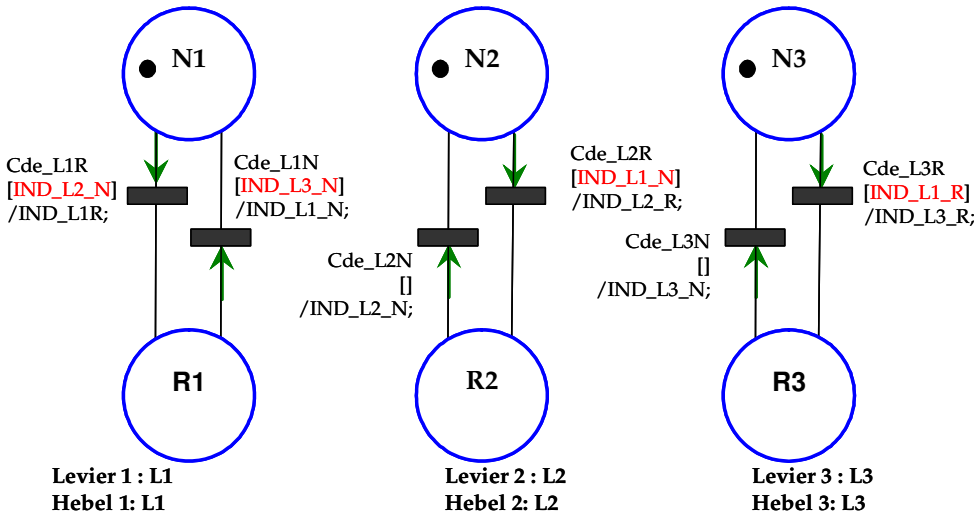



**Figure 7.4 :** Modélisation des positions de trois leviers d'une même table mécanique
**Abbildung 7.4**: Modellierung der Stellungen von drei Hebeln des gleichen Stellwerkes

Les propriétés de sécurité à vérifier (indépendamment de la manière dont le programme fonctionnel a été réalisé) peuvent être directement déduites des incompatibilités recherchées. Les propriétés de sécurité peuvent être traduites comme suit et sous la forme des automates de preuve de la figure 7.5 :

- (L1- L2-)
  ⇒ L1R ET L2R alors «non sûr» ≠ ∅, **(14)**
- (L1+ L3-)
  ⇒ L1N ET L3R alors «non sûr» ≠ ∅,
- (L1- L2-) et L1+ L3-) ⇔ (L2- L3-)
  ⇒ si L2R ET L3R alors «non sûr» ≠ ∅.

Le système décrit n'ayant pas de mémorisation l'automate de preuve est particulièrement simple.

Die zu prüfenden Sicherheitseigenschaften (unabhängig von der Art, in der das funktionelle Programm verwirklicht worden ist), können direkt von den gesuchten Unvereinbarkeiten abgeleitet werden. Die Sicherheitseigenschaften können in Form der Beweisautomaten von Abb. 7.5 umgesetzt werden und auch wie folgt:

- (L1- L2-)
⇒ L1R und L2R dann „unsicher" ≠ ∅, **(14)**
- (L1+ L3-)
⇒ L1N und L3R dann „unsicher" ≠ ∅,
- (L1- L2-) und (L1+ L3-) ⇔ (L2- L3-)
⇒ wenn L2R und L3R dann „unsicher"≠ ∅.

Da das beschriebene System kein Gedächtnis hat, ist der Beweisautomat in Abb.7.5 besonders einfach.

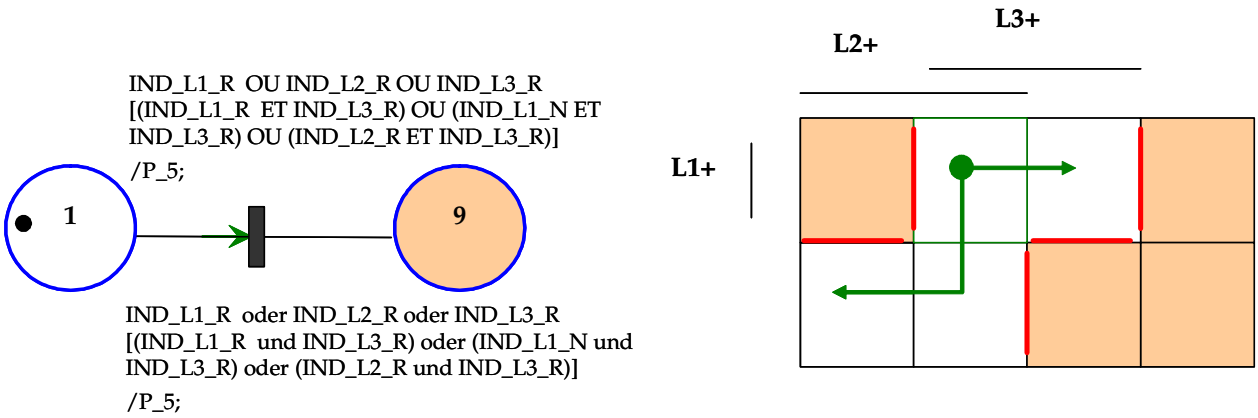



**Figure 7.5** : Automate de preuve
**Abbildung 7.5**: Beweisautomat

L'automate de preuve est évalué après chaque changement d'état d'une entrée externe (en l'occurrence la manœuvre d'un levier). Il vérifie que le traitement de cette n'a violé aucune des propriétés explicitées.

Ainsi, reprenant les termes de la démonstration du Chapitre 6, si jamais la place 9 (indicateur P_5 positionné à «vrai», figure 7.5) n'est atteinte alors que l'ensemble des transitions fonctionnelles possibles entre les états accessibles du système ont été explorés (graphes fonctionnels et graphes de preuve), alors le système est sûr.

L'arbre des états accessibles peut alors être représenté comme suit :

Der Beweisautomat wird nach jeder Änderung des Zustands eines externen Eingangs ausgewertet (im vorliegenden Fall der Bedienung eines Hebels). Er prüft, dass die Bearbeitung der externen Eingabe nicht die ausgedrückten Eigenschaften verletzt hat.

Mithilfe der Begriffe des Beweises aus Kapitel 6 ausgedrückt, kann man sagen, dass das System sicher ist, wenn während der Auswertung aller funktionellen Transitionen zwischen allen erreichbaren Systemzuständen (funktionelle Graphen und Beweisgraphen) der Zustand 9 niemals erreicht wird (Abb.7.5).

Der Baum der erreichbaren Zustände kann wie in Abb. 7.6 darstellt werden.

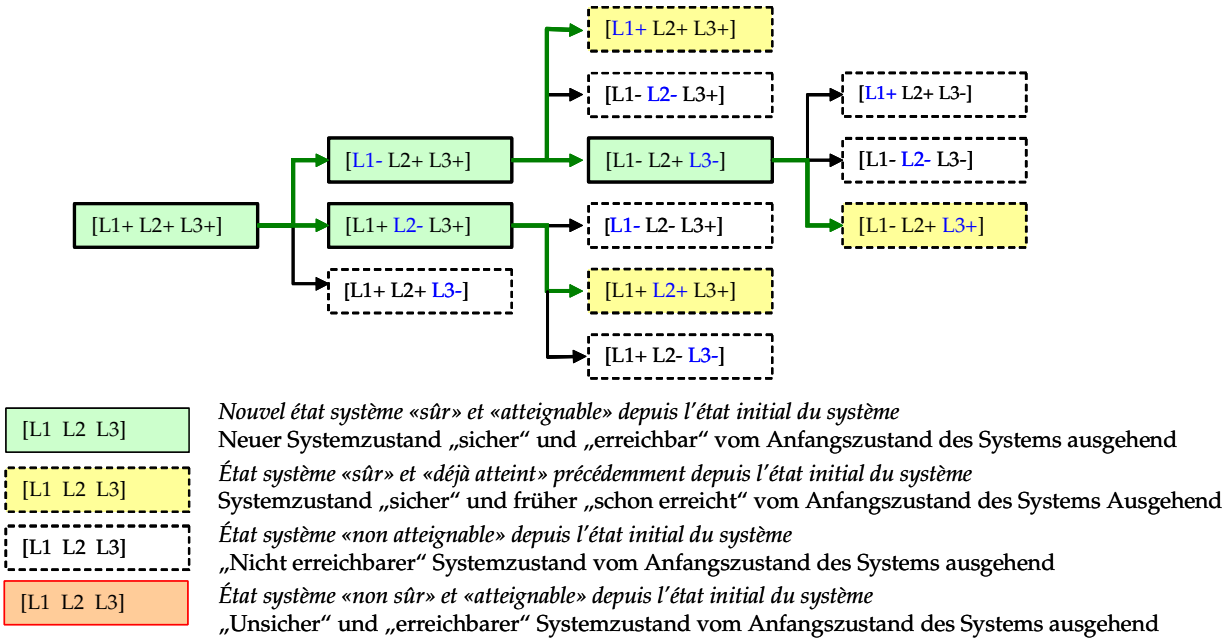



**Figure 7.6 :** L'arbre des états accessibles des trois leviers d'une table mécanique (L1- L2-) (L1+ L3-)
**Abbildung 7.6**: Baum der zugänglichen Zustände der drei Hebel desselben Verschlusstisches (L1- L2-) (L1+ L3-)

Si une erreur a été introduite dans la table d'enclenchement lors de sa réalisation (par exemple l'oubli de mise en place d'un taquet) alors l'arbre obtenu sera différent.

Nous obtiendrions par exemple le modèle fonctionnel suivant : L1R/L3N a été omis (lire : L1 renversé enclenche L3N c'est-à-dire interdit L3R)

Wenn bei der Verwirklichung des Verschlusstisches ein Fehler begangen wurde (zum Beispiel das Vergessen des Setzens eines Verschlussstücks), weist der Verschlusstisch Unterschiede zum erhaltenen Baum auf.

Man würde zum Beispiel (Abb. 7.7) das folgende funktionelle Modell erhalten: L1R/L3N ist vergessen worden (man muss lesen: Hebel L1 umgestellt sichert Hebel L3 normal, verbietet also L3 umgestellt).

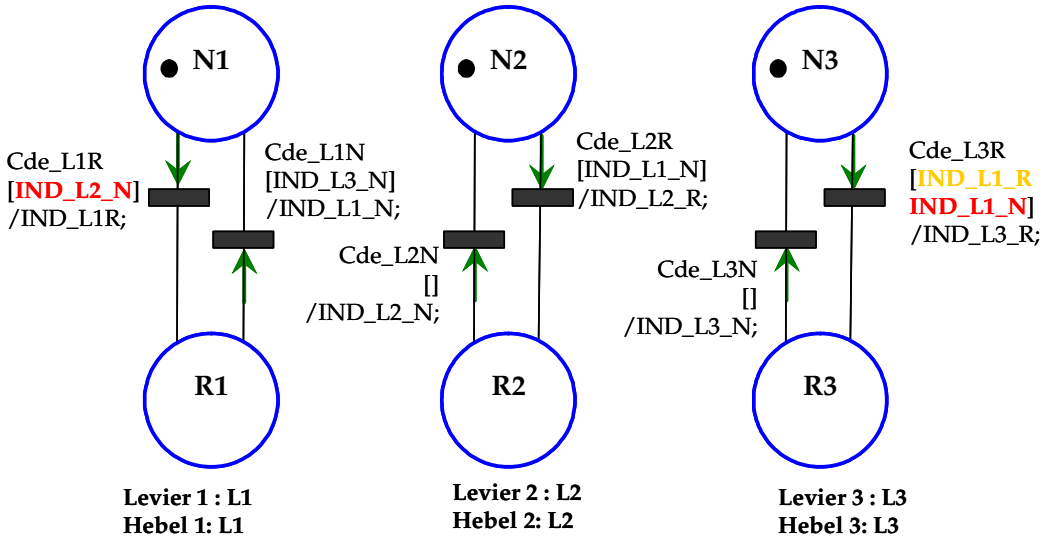



**Figure 7.7 :** Modélisation des positions de trois leviers d'une même table mécanique – avec une erreur de conception
**Abbildung 7.7**: Modellierung der Stellungen von drei Hebeln des gleichen Stellwerkes – mit Konzeptionsfehler

L'exploration des états systèmes et l'interprétation de l'automate de preuve permettent de :

- détecter l'écart entre le fonctionnel réalisé et les propriétés requises : soit l'existence de situation « non sûre », soit l'existence de « surabondant » (non vivacité) ;
- identifier les séquences (chemins) menant aux situations dangereuses.

Die Auswertung der Systemzustände und die Interpretation des Beweisautomaten erlauben:

- die Abweichung zwischen den verwirklichten Funktionen und den erforderlichen Eigenschaften festzustellen: entweder das Vorhandensein von „unsicheren" oder „überflüssigen" Zuständen
- die Sequenzen (Wege) zu identifizieren, die zu gefährlichen Situationen führen.

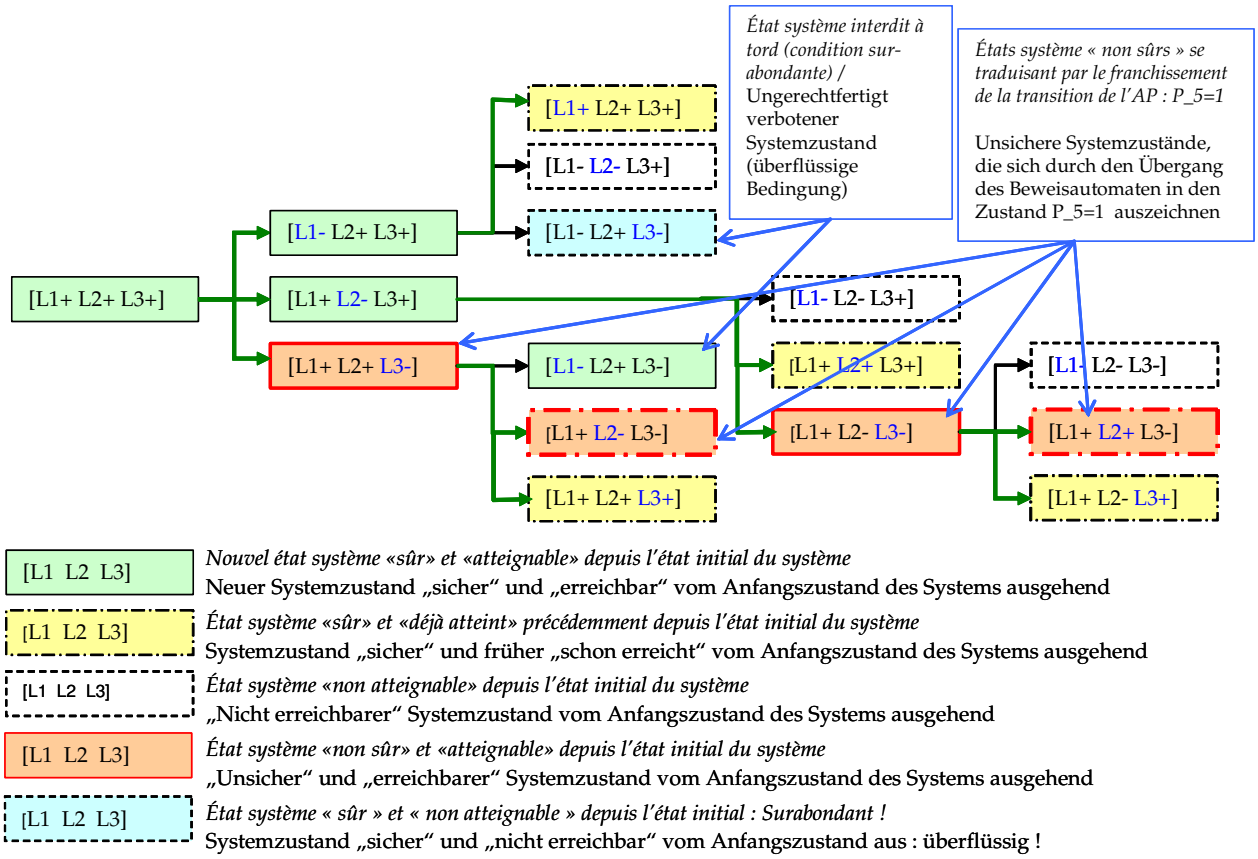



**Figure 7.8 :** L'arbre des états accessibles des trois leviers d'une table mécanique – avec une erreur de conception
**Abbildung 7.8**: Baum der zugänglichen Zustände der drei Hebel eines mechanischen Stelltisches - mit Konzeptionsfehler

Il est remarquable de constater que l'altération d'un des termes de l'incompatibilité donne simultanément naissance :

- à un ou plusieurs surabondants (situation interdite à tord) et à des états non sûrs ;
- à un plus grand nombre d'états système que la situation normale sans défaut (en l'occurrence, 6 au lieu de 4 sur un maximum de 23 = 8 possibles).

Ceci traduit le fait qu'un état système est accessible ou non selon la séquence des changements d'état des entrées.

Cet exemple est simpliste, mais illustre la force de la démarche. Une simple erreur de modélisation se traduit par plusieurs signalements. Les séquences menant aux événements redoutés peuvent ainsi être exhaustivement déterminées :

- Système trop libéral (P_5=1) : {3R} {3R/2R},
- Système trop restrictif : {3R/1R}.

L'absence de changement d'état de l'automate de preuve prouve qu'il n'existe pas d'état système ne respectant pas les propriétés de sécurité ainsi formalisées. La détection états interdits à tord (surabondants) est possible mais requiert l'élaboration d'automates de preuve plus proche du fonctionnel attendu.

### 7.2.1.3 Relations inter poste

Les relations entre postes d'aiguillage, le cas échéant de technologies différentes, constituent un élément clé de la conception des postes d'aiguillage et des installations de signalisation. Les principes suivants ont traversé les technologies. Ils visent à l'assurance qu'à un moment donné les deux leviers (organes de commande) à assujettir ne seront pas activés au même moment dans les deux postes. Ils utilisent une logique à 3 états et requièrent des échanges bidirectionnels entre les deux postes. La figure 7.9 illustre ceci dans le cas d'un enclenchement entre deux postes mécaniques.

Es ist anzumerken, dass die Veränderung von einem der Begriffe der Unvereinbarkeit gleichzeitig:

- ein oder mehrere überflüssige (Situationen, die zu Unrecht verbotenen sind) oder unsichere Situationen erzeugt.
- eine größere Anzahl von Systemzuständen als die normale Situation ohne Fehler (im vorliegenden Fall 6 anstatt 4 von maximal 23 = 8 möglichen) erzeugt.

Dies zeigt, dass ein Systemzustand je nach der Sequenz der Eingabeänderungen erreichbar ist oder nicht. Dieses Beispiel ist einfach, illustriert aber die Stärke der Methode. Ein einfacher Modellierfehler lässt sich durch mehrere Anzeichen erkennen. Die Sequenzen, die zu den befürchteten Ereignissen führen, können so alle bestimmt werden:

- zu freies System (P_5=1): {3R} {3R/2R},
- zu einschränkendes System: {3R/1R}.

Die Abwesenheit einer Zustandsänderung des Beweisautomaten zeigt, dass es keinen Systemzustand gibt, der die Sicherheitseigenschaften nicht respektiert. Das Aufspüren aller fälschlich verbotenen Zustände (überflüssige Zustände) ist möglich, erfordert jedoch den Entwurf von Beweisautomaten, die sich näher an der erwarteten Funktion orientieren.

### 7.2.1.3 Beziehungen zwischen Stellwerken

Die Beziehungen zwischen Stellwerken bilden ein wichtiges Element der Entwicklung der Stellwerke und der Signalisierungseinrichtungen. Die folgenden Grundsätze haben selbst bei Technologiewechsel Bestand. Sie sichern ab, dass zwei zu bedienende Hebel (Bedienorgane) nicht in beiden Stellwerken im gleichen Augenblick umgestellt werden können. Dazu wird eine Logik mit 3 Zuständen benützt, die eine Kommunikation zwischen den beiden Stellwerken in beide Richtungen benötigt. Abb. 7.9 gibt ein Beispiel mit zwei Hebeln.

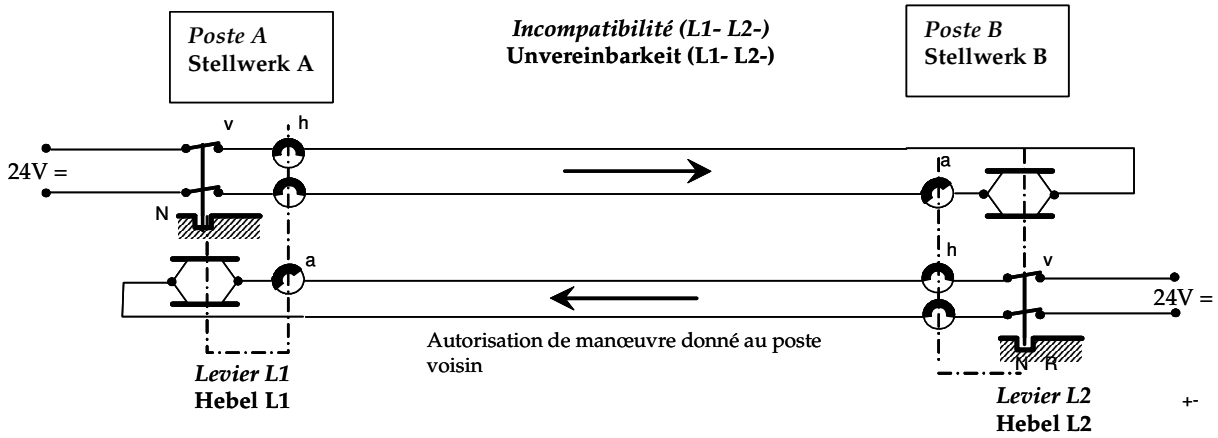



**Figure 7.9a** : Enclenchement binaire entre postes mécaniques
**Abbildung 7.9a**: Binärer Verschluss zwischen zwei mechanischen Stellwerken

Les postulats de fonctionnements suivants doivent être pris en compte pour comprendre le fonctionnement du montage précédent :

- les angulations des contacts électriques et plages de verrouillage du levier (Figure 7.9b) ;

Die folgenden Funktionsanforderungen müssen berücksichtigt werden, um das Funktionieren der vorhergehenden Montage zu begreifen:

- Winkel der elektrischen Kontakte und Bereiche des Hebelverschlusses (Abb. 7.9b)

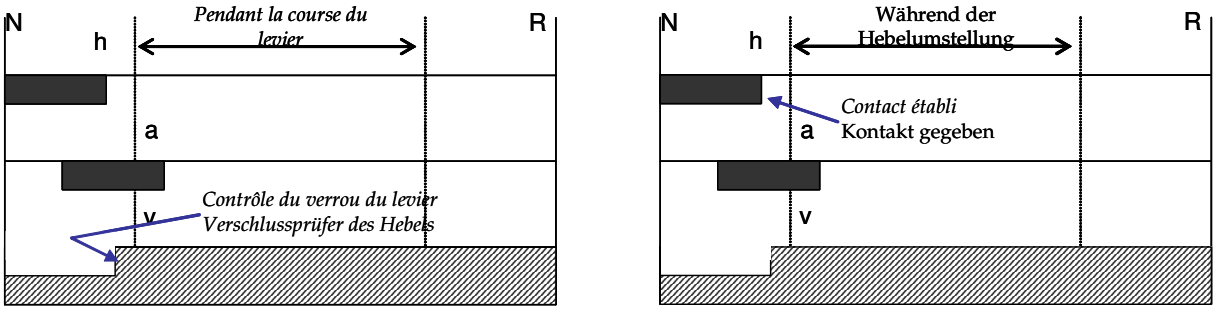



**Figure 7.9b :** Angulations des contacts électriques et plages de verrouillage
**Abbildung 7.9b**: Winkel der elektrischen Kontakte und Bereiche des Verschlusses

- continuité des enclenchements mécaniques et électriques (mécanique avant électrique) ;
- tout incident (la perte d'une communication inter poste) ne doit jamais permettre de s'affranchir de l'incompatibilité (L1- L2-) ;
- la sollicitation simultanée des deux leviers conduit à ce qu'aucun des postes n'ait la priorité.

La modélisation de fonctionnement de l'installation précédente peut être traduite par les graphes suivants :

- Kontinuität der mechanischen und elektrischen Sicherungen (mechanisch vor elektrisch)
- Kein Zwischenfall (der Kommunikationsverlust zwischen den Stellwerken) darf erlauben die Unvereinbarkeit (L1- L2-) aufzulösen.
- Das gleichzeitige Betätigen der zwei Hebel führt dazu, dass kein Stellwerk Vorrang hat.

Die Modellierung der Funktionen der vorigen Anlage kann durch die Graphen der Abb. 7.10 ausgedrückt werden.

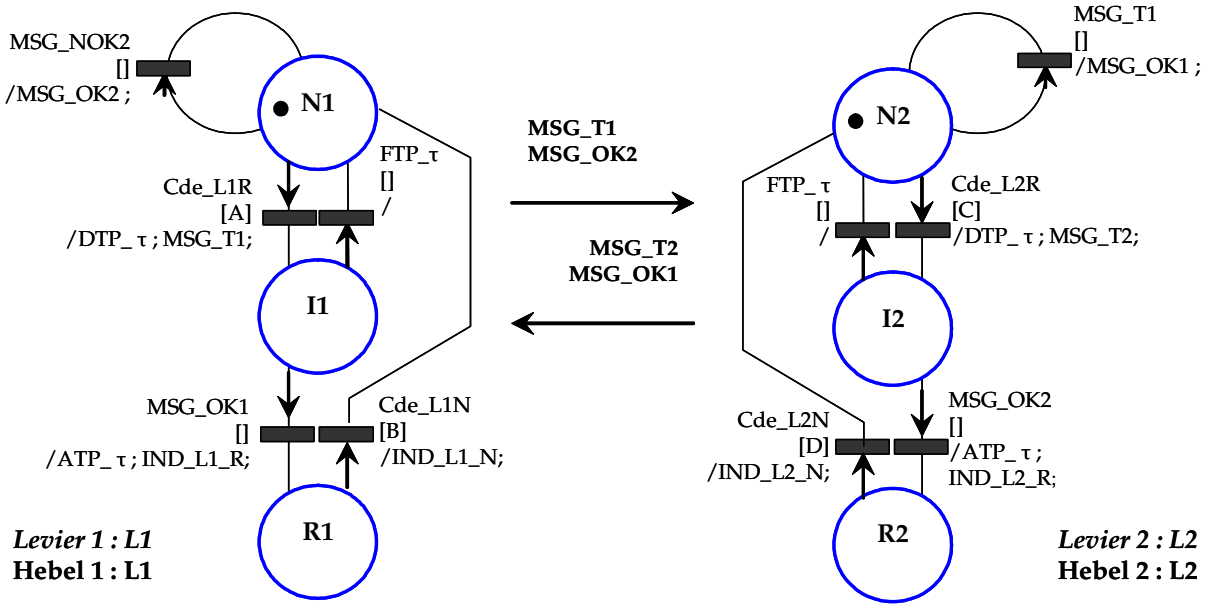



**Figure 7.10 :** Modélisation du principe de fonctionnement d'un enclenchement de sens
**Abbildung 7.10**: Modellierung des Funktionsprinzips eines Richtungsverschlusses

Le vecteur d'état du système peut être défini à l'aide de cinq composantes comme suit :
- $V_1$ : position du levier L1 (N ou R) ou (+ -)
- $V_2$ : test en cours vers poste B (0 ou 1)
- $V_3$ : position du levier L2 (N ou R) ou (+ -)
- $V_4$ : test en cours vers poste A (0 ou 1)
- $V_5$ : position de l'automate de preuve

Der Zustandsvektor des Systems kann mit Hilfe von fünf Komponenten wie folgt definiert werden:
- V1: Lage des Hebels L1 (N oder R) oder (+ -)
- V2: aktueller Test in Richtung des Stellwerks B
- V3: Lage des Hebels L2 (N oder R) oder (+ -)
- V4: aktueller Test in Richtung des Stellwerks A
- V5: Zustand des Beweisautomaten

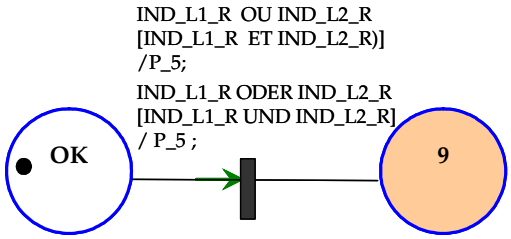



**Figure 7.11 : A**utomate de Preuve / **Abbildung 7.11:** Beweisautomat

**VE initial** = [N 0 N 0 OK] **(15)**

L'exécution de la preuve montre que la fonction est sûre. L'arbre des états accessibles obtenu est illustré par la figure 7.12 suivante :

**Anfänglicher Zustandsvektor** = [N 0 N 0 OK] **(15)**

Die Durchführung des Beweises zeigt, dass die Funktion sicher ist. Der Baum der zugänglichen Zustände ist in Abb. 7.12 wiedergegeben.

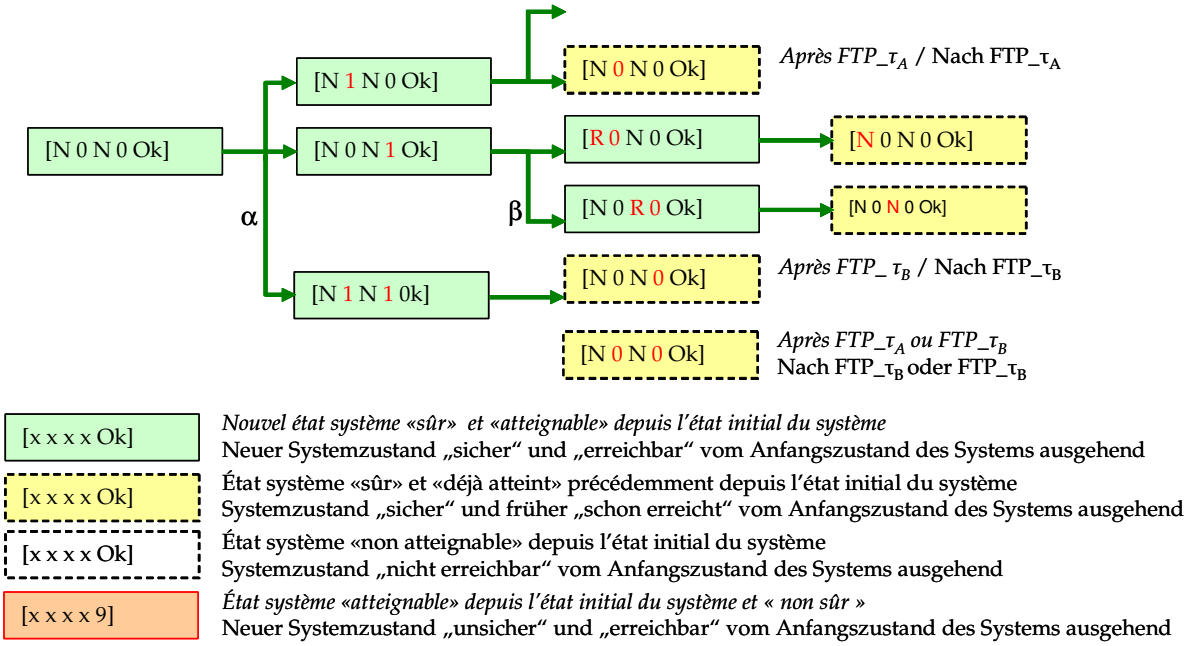



**Figure 7.12:** Arbre des états accessibles – **Abbildung 7.12**: Baum der erreichbaren Zustände

Notons que la sollicitation simultanée des leviers dans chaque poste ou la perte de communications ne conduit jamais à une situation dangereuse.

Les relations entre les postes d'aiguillage (notamment dans le cadre des postes allemands) peuvent aussi reposer sur des relations d'assujettissement d'un poste sur l'autre. La manœuvre d'un levier L1 du poste assujetti n'est alors possible qu'avec l'accord préalable du poste directeur (manœuvre du levier L2).

L'incompatibilité à réaliser est alors : (L1- L2+). Le montage électrique et la modélisation qui en découle sont illustrés par les figures suivantes.

Es ist anzumerken, dass das gleichzeitige Betätigen der Hebel in jedem Stellwerk oder der Kommunikationsverlust niemals zu einer gefährlichen Situation führen kann.

Die Beziehungen zwischen den Stellwerken (besonders bei deutschen Stellwerken) können auch auf einer Hierarchie der Stellwerke untereinander beruhen. Das Manöver eines Hebels L1 des untergeordneten Stellwerks ist dann nur mit der vorherigen Zustimmung des übergeordneten Stellwerks möglich (Manöver des Hebels L2).

Die umzusetzende Unvereinbarkeit ist dann: (L1- L2+). Die elektrische Montage und die daraus abgeleitete Modellierung, werden in den folgenden Abbildungen (7.13. und 7.14) illustriert.

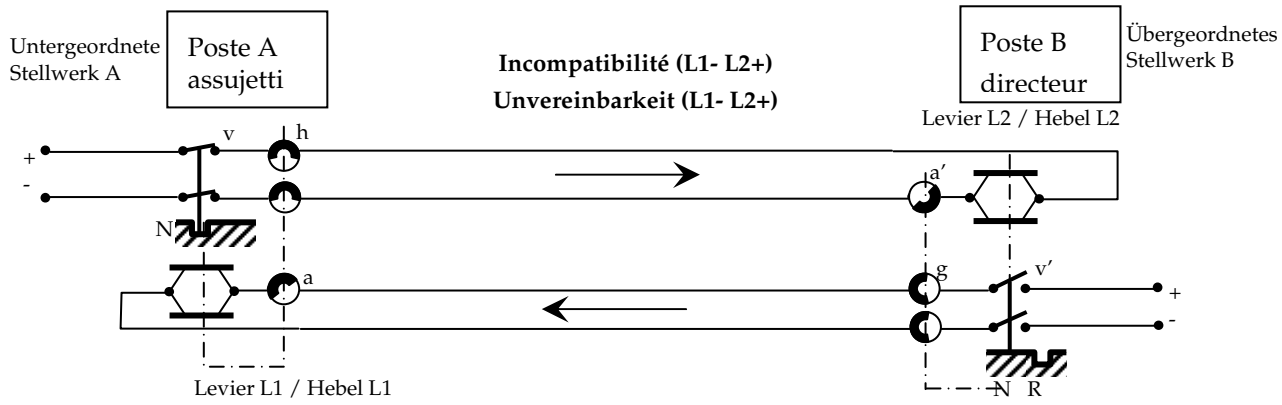



**Figure 7.13** : Enclenchements binaires entre postes mécaniques
**Abbildung 7.13**: Binärer Verschluss zwischen mechanischen Stellwerken

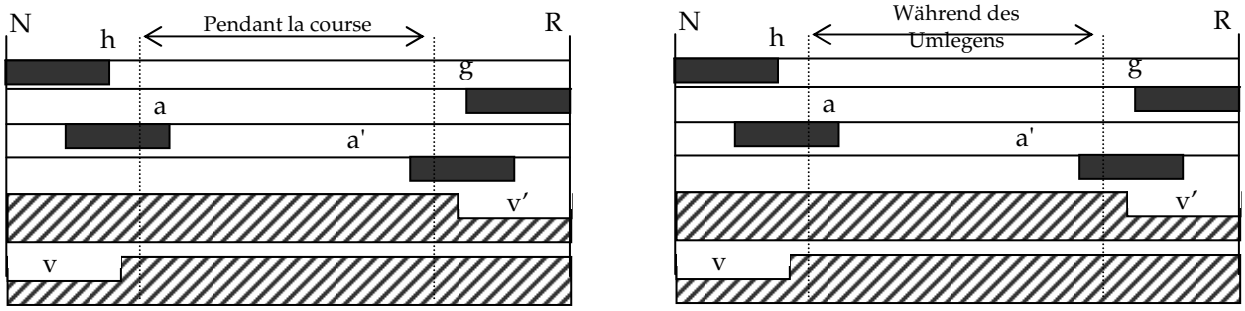



**Figure 7.14** : Angulations des contacts électriques et plages de verrouillage
**Abbildung 7.14**: Winkel der elektrische Kontakte und Bereiche des Verschlusses

La modélisation de l'installation précédente peut être traduite par les graphes suivants :

Die Modellierung der Anlage kann durch die Graphen der Abbildung 7.15 dargestellt werden.

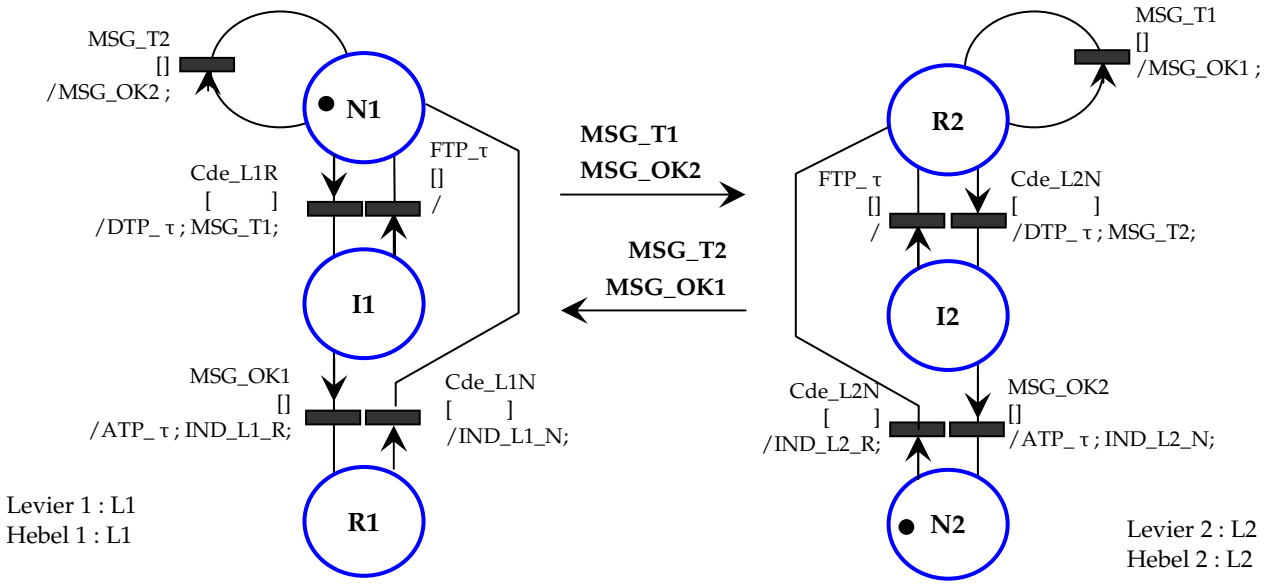



**Figure 7.15** : Enclenchements binaire entre postes mécaniques - Modélisation
**Abbildung 7.15**: Binärer Verschluss zwischen zwei mechanischen Stellwerken - Modellierung

L'automate de preuve, correspondant à l'incompatibilité (L1- L2+), est illustré par la figure 7.16.

Der Beweisautomat, der der Unvereinbarkeit (L1- L2+) entspricht, sieht dann so wie in Abbildung 7.16 aus.

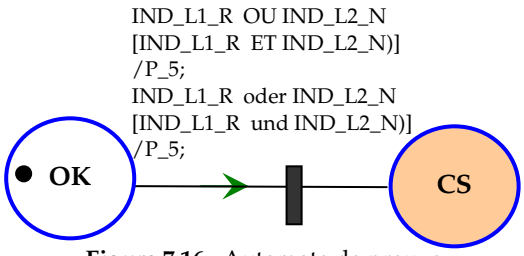



**Figure 7.16** : Automate de preuve
**Abbildung 7.16** : Beweisautomat

La démarche précédente conduit de même à prouver que le système est sûr.

Das vorangehende Verfahren beweist ebenso, dass das System sicher ist.

### 7.2.1.4 Échanges avec temporisation

Il peut paraître tentant pour les concepteurs de réaliser ce type de communication entre modules informatiques d'enclenchement au moyen d'une simple relation unidirectionnelle. Cette simplification est une violation de ce principe centenaire, elle utilise les délais. [cf. lien RBC => ERTMS] Une telle approche peut fonctionner en situation normale. En situation dégradée, nous allons montrer qu'il existe des possibilités d'évolution des entrées telles que les propriétés de sécurité ne sont plus respectées. Le modèle fonctionnel correspond à la réalisation de l'incompatibilité (L2+ L1-) [avec l'automatisme Cde L2R ⇒ L2R ⇒ L1R, puis Cde L2N ⇒ L1N ⇒ L2N].

### 7.2.1.4 Austausch mit Verzögerung

Es kann für die Entwickler verlockend sein, diesen Kommunikationstyp zwischen Informatiksicherungsmodulen mithilfe einer einfachen einseitigen Beziehung durchführen. Diese Vereinfachung ist eine Verletzung eines hundertjährigen Grundsatzes, da sie Verzögerungen benützt [vgl. RBC=>ERTMS].
Ein solches Konzept kann im normalen Zustand funktionieren. In der Rückfallebene kann man zeigen, dass es Möglichkeiten der Eingabeveränderungen gibt, die dazu führen, dass die Sicherheitseigenschaften nicht mehr respektiert werden. Das funktionelle Modell entspricht der Verwirklichung der Unvereinbarkeit (L2+ L1-) [mit der Regelung Cde L2R ⇒ L2R ⇒ L1R dann Cde L2N ⇒ L1N ⇒ L2N].

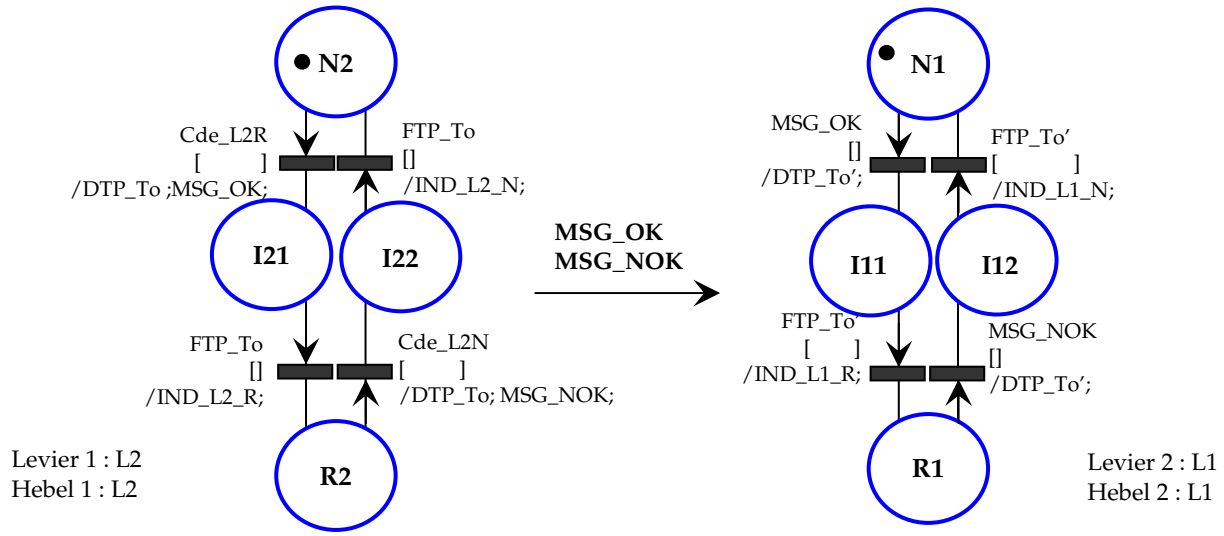



**Figure 7.17** : Enclenchements binaire entre postes mécaniques – avec temporisation
**Abbildung 7.17**: Binärerverschluss zwischen mechanischen Stellwerken – mit Verzögerung

L'automate, correspondant à l'incompatibilité (L1- L2+), est celui de la figure 7.17. La démarche de preuve montre que le système est n'est pas sûr (P_5 = 1). En effet, les séquences suivantes conduisent à un état non sûr :

- Cde_L2R→FTP_To (si To < To' ou si perte de Msg)
- Cde_L2N→FTP_To (si To < To' ou si perte de Msg)

Der Automat, der der Unvereinbarkeit (L1- L2 +) entspricht, ist in Abbildung 7.17 zu sehen. Der Beweislauf zeigt, dass das System nicht sicher ist (P_5 = 1). In der Tat führen die folgenden Sequenzen zu einem unsicheren Zustand:

- Cde_L2R→FTP_To (wenn To < To' oder bei Verlust der Nachricht)
- Cde_L2N→FTP_To (wenn To < To' oder bei Verlust der Nachricht)

## 7.2.2 Cas d'une bifurcation simple

Étendons notre démarche au cas d'une bifurcation commandée par un poste mécanique. Pour ce faire, il s'agit de modéliser comme précédemment la table mécanique et les enclenchements électriques d'aiguille, puis écrire les automates de preuve vérifiant la correction de notre modélisation.

Cette fois-ci le modèle couvrira l'ensemble des enclenchements du poste, mécaniques et électriques, notamment ceux liés au block permissif en vigueur en France (FA. AEAP).

### 7.2.2.1 Présentation du plan de voie et du programme du poste

Le plan de voie de la bifurcation est illustré par la figure suivante :

## 7.2.2 Fall einer Abzweigung

Die neu entwickelte Methode wird nun erweitert und auf den Fall einer von einem mechanischen Stellwerk gesteuerten Abzweigung angewandt. Um dies durchzuführen modelliert man wie zuvor den mechanischen Stelltisch und die elektrischen Verschlüsse der Weiche. Danach werden die Beweisautomaten formuliert, die die Erweiterung der Modellierung prüfen.

In diesem Fall deckt das Modell alle mechanischen und elektrischen Verschlüsse des Stellwerks, insbesondere jene, die mit dem in Frankreich verwendeten Block mit bedingten Haltsignalen zusammenhängen (FA. AEAP), ab.

### 7.2.2.1 Gleisplan und Stellwerksprogramm

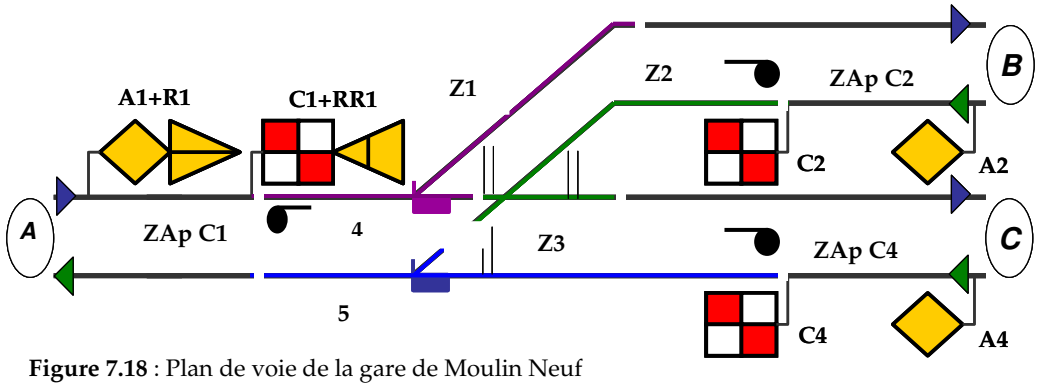



**Figure 7.18** : Plan de voie de la gare de Moulin Neuf
**Abbildung 7.18**: Gleisplan des Bahnhofs Moulin Neuf

| | | | |
|---|---|---|---|
| **L1** | A1 | **L5** | Aig 5 |
| **L2** | C1 vers B<br>R1+RR1 | **L6** | C2 vers A |
| | | **L7** | A2 |
| **L3** | C1 vers C | **L8** | C4 vers A |
| **L4** | Aig 4 | **L9** | A4 |

Affectation des entrées terrain :

| | | | |
|---|---|---|---|
| **Z1** | **ZApC1** | **PdC1** | **KAg4G** |
| **Z2** | **ZApC2** | **PdC2** | **KAg4D** |
| **Z3** | **ZApC4** | **PdC4** | **KAg5G** |
| | | | **KAg5D** |

**Tableau 7.1** : Affectation des leviers de manœuvre du poste

Les signaux sont équipés d'une part d'un dispositif de fermeture automatique (FA par Zone et Pd) et, d'autre part, d'une annulation de l'enclenchement d'approche (AEAP) permettant de rouvrir rapidement le signal derrière une circulation si l'itinéraire est conservé[77].

| | | | |
|---|---|---|---|
| **L1** | A1 | **L5** | Weiche 5 |
| **L2** | C1 hin zu B<br>R1+RR1 | **L6** | C2 hin zu A |
| | | **L7** | A2 |
| **L3** | C1 hin zu C | **L8** | C4 hin zu A |
| **L4** | Weiche 4 | **L9** | A4 |

Verwendung der externen Eingänge:

| | | | |
|---|---|---|---|
| **Z1** | **ZApC1** | **PdC1** | **KAg4G** |
| **Z2** | **ZApC2** | **PdC2** | **KAg4D** |
| **Z3** | **ZApC4** | **PdC4** | **KAg5G** |
| | | | **KAg5D** |

**Tabelle 7.1**: Verwendung der Stellhebel des Stellwerkes

Abb. 7.18 zeigt den Gleisplan mit der Abzweigung. Die Signale sind einerseits mit einer automatischen Schließvorrichtung (FA durch Gleisstromkreise und mechanische Gleisschaltmittel) und andererseits mit einer Annullierung des Annäherungsverschlusses (AEAP) ausgestattet, der es erlaubt, das Signal hinter einem Zug schnell wieder zu öffnen, wenn die Fahrstraße bestehen bleibt.

[77] Disposition spécifique aux postes d'aiguillage français

### 7.2.2.2 Fonctions du poste d'aiguillage

Les incompatibilités identifiées selon les méthodes classiques sont les suivantes :

**Aiguilles :**

Mécanique : **(5- 4+)**

Électrique : **(4↕ Z1-) (5↕ Z2-) (5↕ Z3-)** **(16)**

**Signaux Impairs :**

Mécanique : **(1- 2+ 3+) (2- 4+) (3- 4-)**

Électrique : **(2↑ ZApC1- AEAPC1-)**
**(3↑ ZApC1- AEAPC1-)** **(17)**

**Signaux Pairs :**

Mécanique : **(7- 6+) (6- 5+) (9- 8+) (8- 5-)**

Électrique : **(6↑ ZApC2- AEAPC2-)**
**(8↑ ZApC4- AEAPC4-)** **(18)**

Notons qu'il existe un parallèle direct entre les incompatibilités mécaniques et les enclenchements réalisés par la table mécanique (mise en place des taquets solidaires des leviers du poste) [Descubes, 1898].

**Tableau des résultants :**

| L | N (normal) | R (renversé) | PC |
|---|---|---|---|
| 1 | | (2+3+) / (3+4+) (2+4-) | |
| 2 | (1-3+) / (1-4-) (1-5-) | 4+ / 3- | 1- |
| 3 | (1-2+) / (1-4+) | 4- / 2- 5- | 1- |
| 4 | 2- 5- / | 3- / 6- | |
| 5 | 6- / | 4+ 8- / 3- 9- | |
| 6 | 7- / | 5+ / 4+ 8- | |
| 7 | | 6+ / | |
| 8 | 9- / | 5- / 6- | |
| 9 | | 8+ / 5- | |

**Tableau 7.2** : Enclenchements résultats

Le tableau 7.2 fait apparaître les incompatibilités résultantes crées (elles sont positionnées à droite du signe « / »). Ces incompatibilités, si elles sont utiles, ne nécessitent pas de réalisation physique supplémentaire pour exister. Les incompatibilités superflues ne doivent pas empêcher l'exploitation normale de poste comme illustré dans le tableau des parcours suivant :

Tableau des parcours :

| Iti | Ordre de manœuvre des leviers |
|---|---|
| A→B | L4R - L2R - L1R |
| A→C | 3R – 1R |
| B→A | 8R – 9R |
| C→A | 4R – 5R – 6R – 7R |

**Tableau 7.3** : Tableau des parcours

### 7.2.2.2 Stellwerksfunktion

Die Unvereinbarkeiten, die nach den klassischen Methoden identifiziert wurden, sind folgende:

**Weiche:**

Mechanisch: **(5- 4+)**

Elektrisch: **(4↕ Z1-) (5↕ Z2-) (5↕ Z3-)** **(16)**

**Ungerade Signale:**

Mechanisch: **(1- 2+ 3+) (2- 4+) (3- 4-)**

Elektrisch: **(2↑ ZApC1- AEAPC1-)**
**(3↑ ZApC1- AEAPC1-)** **(17)**

**Gerade Signale:**

Mechanisch: **(7- 6+) (6- 5+) (9- 8+) (8- 5-)**

Elektrisch: **(6↑ ZApC2- AEAPC2-)**
**(8↑ ZApC4- AEAPC4-)** **(18)**

Es ist anzumerken dass es eine Parallele zwischen den mechanischen Unvereinbarkeiten und dem Verschluss gibt, der durch das mechanische Register umgesetzt wird (zusammenhängende Verschlussstücke für die Hebel des Stellwerks) [Descubes, 1898].

**Tabelle der Ergebnisse:**

| L | N (Gerade) | R (Umgestellt) | PC |
|---|---|---|---|
| 1 | | (2+3+) / (3+4+) (2+4-) | |
| 2 | (1-3+) / (1-4-) (1-5-) | 4+ / 3- | 1- |
| 3 | (1-2+) / (1-4+) | 4- / 2- 5- | 1- |
| 4 | 2- 5- / | 3- / 6- | |
| 5 | 6- / | 4+ 8- / 3- 9- | |
| 6 | 7- / | 5+ / 4+ 8- | |
| 7 | | 6+ / | |
| 8 | 9- / | 5- / 6- | |
| 9 | | 8+ / 5- | |

**Tabelle 7.2:** Tabelle der Ergebnisse

Tabelle 7.2 fasst die Ergebnisse der Unverträglichkeiten zusammen (sie befinden sich rechts vom Zeichen „/“). Diese Unvereinbarkeiten erfordern, wenn sie sinnvoll sind, keine weiteren physikalischen Umsetzungen. Die überflüssigen Unvereinbarkeiten dürfen den normalen Stellwerksbetrieb, so wie er in der Streckentabelle (Tab. 7.3) dargestellt ist, nicht behindern.

Strecken Tabelle:

| Fahrstraße | Reihenfolge der Bedienung der Hebel |
|---|---|
| A→B | L4R - L2R - L1R |
| A→C | 3R – 1R |
| B→A | 8R – 9R |
| C→A | 4R – 5R – 6R – 7R |

**Tabelle 7.3:** Streckentabelle

### 7.2.2.3 Modélisation du poste

La modélisation du poste sous forme de graphes conduits au modèle suivant (Figure 7.19) :

### 7.2.2.3 Stellwerksmodellierung

Die Beschreibung des Stellwerkes von Moulin-Neuf in Form von Graphen führt zu dem Modell in Abbildung 7.19.

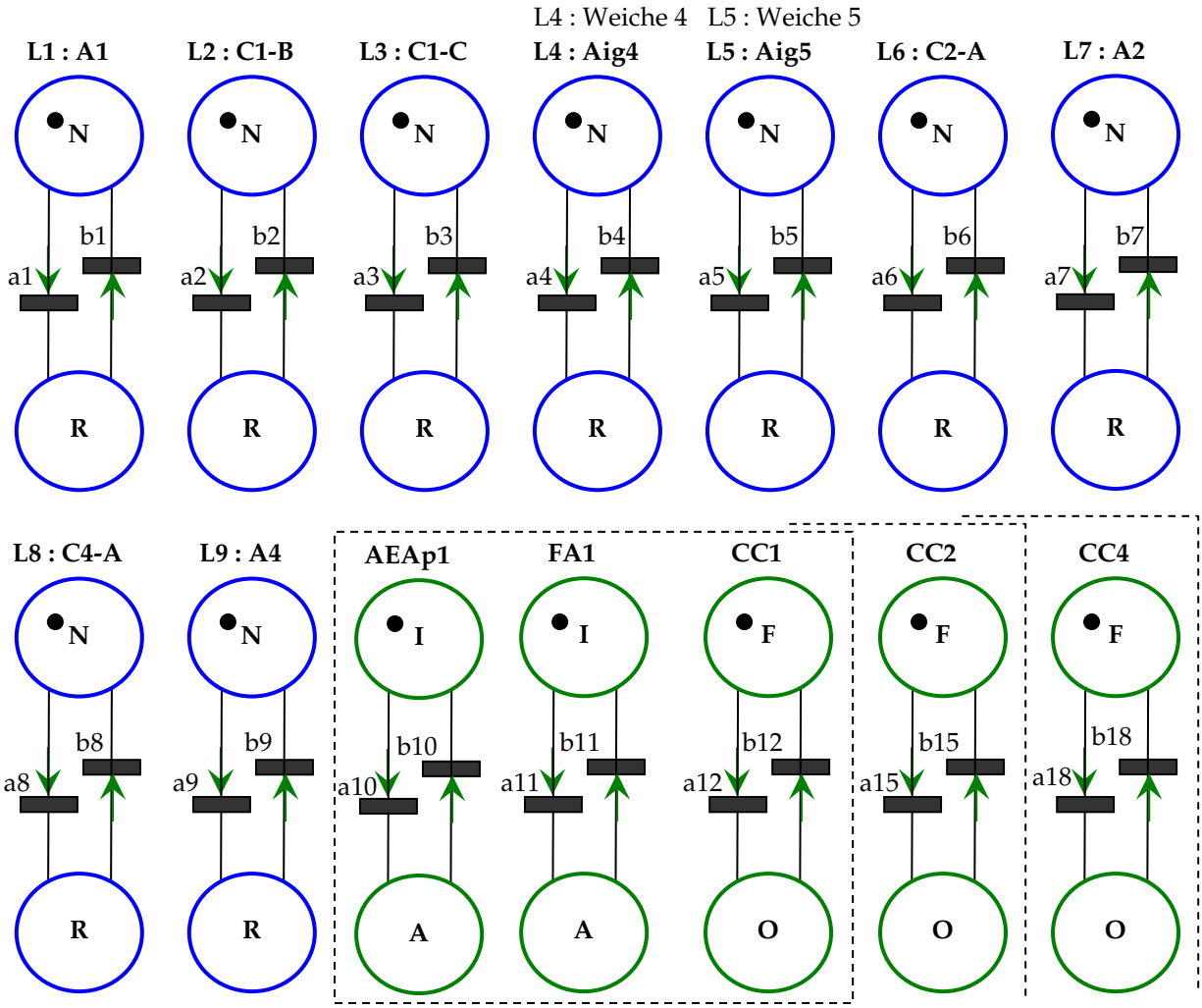



**Figure 7.19** : Modélisation du poste d'aiguillage en langage AEFD
**Abbildung 7.19**: Stellwerksmodelierung in der AEFD-Sprache

Position des leviers :

N = position normale / R = position renversée

Position des dispositifs électriques :

I = enclenchement inactif / A = enclenchement actif

Position des fonctions de contrôle :

F = signal fermé / O = signal ouvert

Position der Hebel:

N = normale/R = umgestellte Position

Position der elektrischen Vorrichtungen:

I = inaktiver Verschluss /A = aktiver Verschluss

Position der Kontrollfunktionen:

F = geschlossenes Signal /O = offenes Signal

| | Événement / Ereignis | Condition / Bedingung | Action / Handlung |
|---|---|---|---|
| a1 | Cde_A1_R | [2R ou 3R] | /IND_CA1_O; |
| b1 | Cde_A1_N | | /IND_CA1_F; |
| a2 | Cde_C1_R-B | [4R] | / |
| b2 | Cde_C1_N-B | [1N et (CTL_ZApC1+ ou IND_AEFd1_A)] | / |
| a3 | Cde_C1_R-C | [4N et 5N] | / |
| b3 | Cde_C1_N-C | [1N et (CTL_ZApC1+ ou IND_AEFd1_A)] | / |
| a4 | Cde_Aig4_G (Weiche 4 Links) | [CTL_Z1+] | /IND_AG4_G; |
| b4 | Cde_Aig4_D (Weiche 4 Rechts) | [5N et CTL_Z1+] | /IND_AG4_D; |
| a5 | Cde_Aig5_G (Weiche 5 Links) | [4N et 8R et CTL_Z2+ et CTL_Z3+] | /IND_AG5_G; |
| b5 | Cde_Aig5_D (Weiche 5 Rechts) | [6R et CTL_Z2+ et CTL_Z3+] | /IND_AG5_D; |
| a6 | Cde_C2_R-A | [5R] | / |
| b6 | Cde_C2_N-A | [7R et (CTL_ZApC2+ et IND_AEFd2_A)] | / |
| a7 | Cde_A2_R | [2R ou 3R] | /IND_CA2_O; |
| b7 | Cde_A2_N | | /IND_CA2_F; |
| a8 | Cde_C4_R-A | [5N] | / |
| b8 | Cde_C4_N-A | [9R et (CTL_ZApC4+ et IND_AEFd4_A)] | / |
| a9 | Cde_A4_R | [2R ou 3R] | /IND_CA4_O; |
| b9 | Cde_A4_N | | /IND_CA4_F; |

**Tableau 7.4:** Tableau des résultants

*NB : ou = oder / et = und*

**Tabelle 7.4:** Tabelle der Ereignisse

Fermeture automatique (FA) des signaux carrés (électriquement) le levier restant en position renversée (les aiguilles de l'itinéraire restent immobilisées par les enclenchements mécaniques) :

Automatische Schließung (FA) der Hauptsignale (elektrisch); der Hebel bleibt in der umgestellten Position (die Weichen der Strecke bleiben durch mechanische Verschlüsse blockiert) (Tab. 7.5).

| | Événement / Ereignis | Condition / Bedingung | Action / Handlung |
|---|---|---|---|
| a10 | CTL_Z1- ou CTL_Pd1- | [CTL_Z1- et CTL_Pd1-] | / IND_FA1_A; |
| b10 | 2N ou 2N | [2N et 3N] | / IND_FA1_I; |
| a13 | CTL_Z2- ou CTL_Pd2 | [CTL_Z3- et CTL_Pd2+] | / IND_FA2_A; |
| b13 | 6N | [6N] | / IND_FA2_I; |
| a16 | CTL_Z3- ou CTL_Pd4 | [CTL_Z2- et CTL_Pd4-] | / IND_FA4_A; |
| b16 | 8N | [8N] | / IND_FA4_I; |

**Tableau 7.5:** Fermeture automatique

*NB : ou = oder, et = und*

**Tabelle 7.5:** Automatische Schließung

Annulation de l'enclenchement d'approche permettant la confirmation de la fermeture du signal carré pour une éventuelle réouverture du même signal pour la circulation suivante abordant le signal à vitesse limitée du fait du block permissif (Les aiguilles de l'itinéraire restent immobilisée au moyen de l'action des circuits de voie et des enclenchements de transit, autorisant ainsi une libération progressive de l'itinéraire à mesure de l'avancement de la circulation ferroviaire. Les leviers d'aiguille sont en effet mécaniquement libres dès la confirmation de la fermeture du signal carré).

Die Annullierung des Annäherungsverschlusses erlaubt die Bestätigung des Schließens des Hauptsignals um dieses Signal für einen nachfolgenden Zug eventuell wieder zu öffnen, der aufgrund des permissiven Blocks mit einer begrenzten Geschwindigkeit fährt. (Die Weichen der Fahrstraße bleiben aufgrund der Einwirkung der Gleisstromkreise und der Fahrstraßenverschlüsse blockiert, was in Abhängigkeit von der Position der Züge nach und nach eine Auflösung der Fahrstraße erlaubt. Die Hebel der Weiche sind mechanisch frei sobald das Schließen des Hauptsignals bestätigt wurde).

| | Événement / Ereignis | Condition / Bedingung | Action / Handlung |
|---|---|---|---|
| a11 | CTL_ZAp1- ou IND_FA1_A ou 2R ou 3R | [CTL_ZAp1- et IND_FA1_A et (2R ou 3R)] | / IND_AEFd1_A; |
| b11 | 2N ou 3N | | / IND_AEFd1_I; |
| a14 | CTL_ZAp2- ou IND_FA2_A ou 6R | [CTL_ZAp2- et IND_FA2_A et 6R] | / IND_AEFd2_A; |
| b14 | 6N | | / IND_AEFd2_I; |
| a17 | CTL_ZAp4- ou IND_FA4_A ou 8R | [CTL_ZAp4- et IND_FA4-A et 8R] | / IND_AEFd4_A; |
| b17 | 8N | | / IND_AEFd4_I; |

**Tableau 7.6 :** Annulation de l'enclenchement d'approche

**Tabelle 7.6:** Annullierung des Annäherungsverschlusses

La commande du signal carré (CC1) assure le contrôle impératif permanent des conditions de sécurité de l'itinéraire pour lequel le signal est ouvert. (Positions des organes de commande, position des appareils commandés, position des capteurs utiles).

Die Steuerung des Hauptsignals (CC1) (vgl. Tab. 7.7) erlaubt eine ständige und unabdingbare Kontrolle der Sicherheitsbedingungen der Fahrtstraße, für die das Signal geöffnet wird (Lage der Bedienorgane, Lage der Steuergeräte, Lage der benötigten Sensoren).

| | Événement / Ereignis | Condition / Bedingung | Action / Handlung |
|---|---|---|---|
| a12 | CTL_K4G+ ou CTL_K4D+ ou 2R ou 3R ou IND_FA1_I | [((CTL_K4G+ et 2R) ou (CTL_K4D+ et 3R)) et IND_FA1_I] | / IND_AEFd1_A; |
| b12 | CTL_K4G- ou CTL_K4D- ou 2N ou 3N ou IND_FA1_A | [non ((CTL_K4G+ et 2R) ou (CTL_K4D+ et 3R)) et IND_FA1_I] | / IND_AEFd1_I; |
| a15 | CTL_K5G+ ou 6R ou IND_FA2_I | [(CTL_K5G+ et 6R) et IND_FA2_I] | / IND_AEFd2_A; |
| b15 | CTL_K5G- ou 6N ou IND_FA2_A | [non (CTL_K5G+ et 6R) et IND_FA2_I] | / IND_AEFd2_I; |
| a18 | CTL_K5D+ ou 8R ou IND_FA4_I | [(CTL_K5D+ et 8R) et IND_FA4_I] | / IND_AEFd4_A; |
| b18 | CTL_K5D- ou 8N ou | [non (CTL_K5D+ et 8R) et IND_FA4_I] | / IND_AEFd4_I; |

**Tableau 7.7 :** Commande du signal carré C1

**Tabelle 7.7:** Bedienung der Hauptsignals C1

La position du système peut être résumée au travers du vecteur d'état suivant :

Die Position des Systems kann durch den folgenden Zustandsvektor zusammengefasst werden:

**VE = [L1 à L9 / FA1 à FA4 / AEAp1 à 4 / CC1 à 4 / CTL1 à N]**

**Zustandsvektor VE = [L1 bis L9 / FA1 bis FA4 / AEAp1 bis 4 / CC1 bis 4 / CTL1 bis N]** **(19)**

Le vecteur d'état initial correspond à la situation où tous les leviers sont en position normale (N) et l'ensemble des automatismes électriques sous tension (I), les contrôles terrains en cohérence avec les organes de commande (levier). Les entrées terrains celles définies par les tableaux 7.1 et 7.2.

Der anfängliche Zustandsvektor entspricht der Lage, in der alle Hebel in normaler Position (N) sind, alle elektrischen Steuerungen unter Spannung stehen (I) und die Kontrollen vor Ort den Bedienorganen (Hebeln) entsprechen. Die Eingänge vor Ort entsprechen denen der Tab. 7.1 und 7.2.

**VE = [1N 2N 3N 4N 5N 6N 7N 8N 9N I1 I2 I4 I1 I2 I4 F1 F2 F4 / état des CTL +/-]**

**Zustandsvektor = [1N 2N 3N 4N 5N 6N 7N 8N 9N I1 I2 I4 I1 I2 I4 F1 F2 F4 / Zustand der CTL +/-]** **(20)**

La présente modélisation peut être explorée exhaustivement

Diese Modellierung kann erschöpfend analysiert werden.

Les résultats détaillés sont trop longs pour être inscrits ici exhaustivement [état systèmes distincts]. A titre d'illustration nous présentons ci après l'arbre des états systèmes décrivant :

1. le franchissement de l'itinéraire A vers B par une circulation ;
2. avec réouverture au plus tôt pour un train suivant l'itinéraire A vers B.

Die detaillierten Ergebnisse [alle verschiedenen Systemzustände] sind zu lang, um sie hier aufzuführen. Zur Veranschaulichung wird nachfolgend der Baum der Systemzustände aufgeführt, der:

1. die Überfahrung der Fahrstraße von A nach B durch einen Zug beschreibt,
2. die Fahrstraße von A nach B für den nächsten Zug frühstmöglich öffnet.

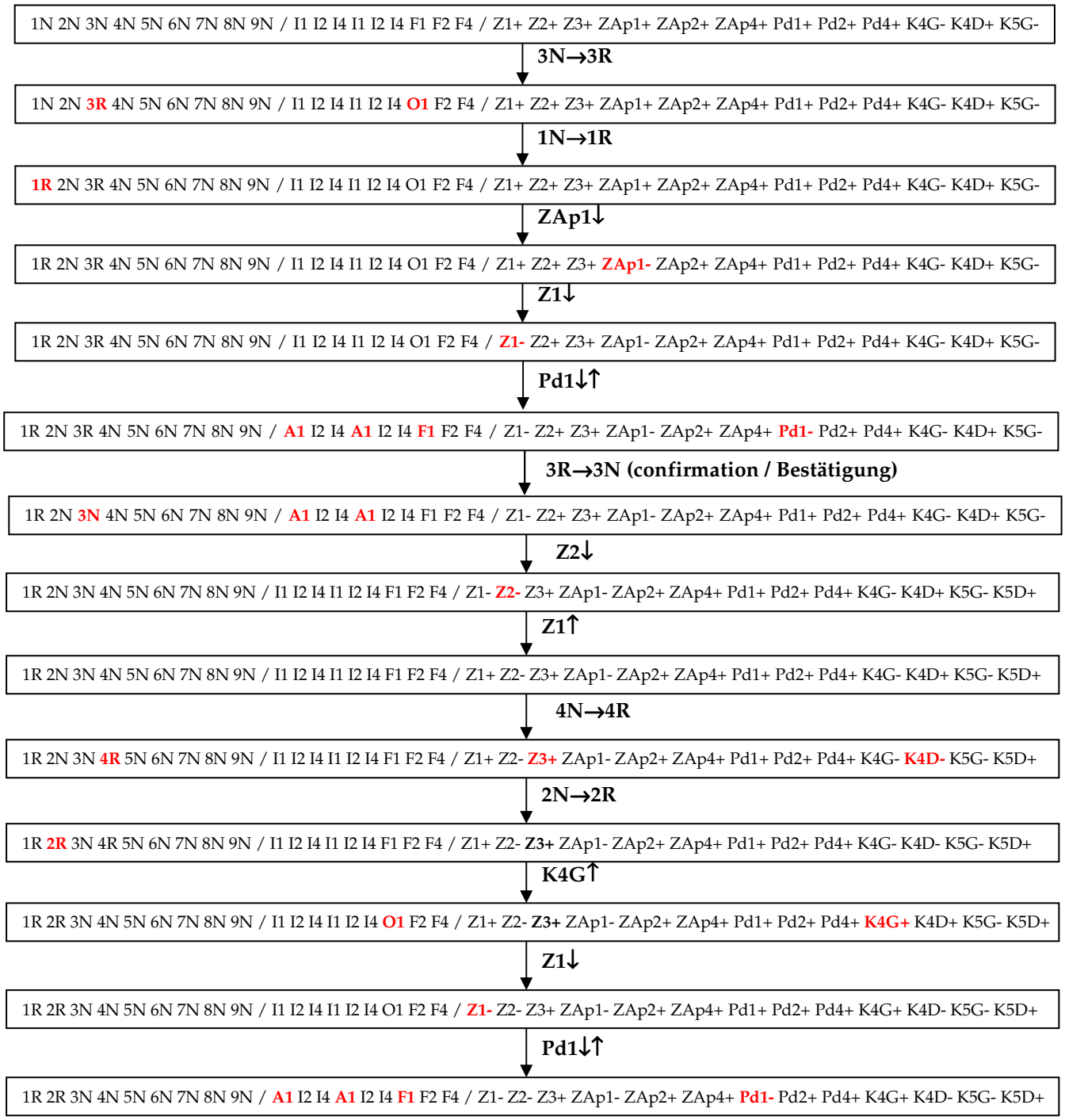



**Figure 7.20** : Succession d'états système correspondants au passage successif de deux circulations
**Abbildung 7.20**: Folge der Systemzustände, die dem Vorbeifahren von zwei sich folgenden Zügen entsprechen

Le fonctionnel, même simplifié, assure une fonction de transit souple (cf. Chapitre 4). L'exploration permet ainsi de retrouver les situations limites. Cette recherche de souplesse maximale repose sur un transfert des enclenchements mécaniques de formation (3R impose 4N, 6N, 7N) sur des enclenchements électriques de transit qui gèrent le dégagement de l'itinéraire par la circulation, libérant ainsi la table mécanique pour un autre itinéraire de même origine (ou utilisant les mêmes ressources).

Selbst vereinfacht, erfüllt die Funktion eine abschnittsweise Fahrstraßenauflösung (siehe Kapitel 4). Die Auswertung erlaubt es so, Grenzsituationen zu identifizieren. Dieser Versuch, die größtmögliche Flexibilität zu erzielen, beruht auf der Übertragung der mechanischen Fahrstraßenbildungsverschlüsse (3R verlangt 4N, 6N, 7N) auf die elektrischen Fahrstraßenverschlüsse, bei denen die Fahrstraße durch den Zug aufgelöst wird und so die mechanische Stelltafel für eine andere Fahrstraße gleichen Ursprungs (oder mit der gleichen Ressourcennutzung) freigibt.

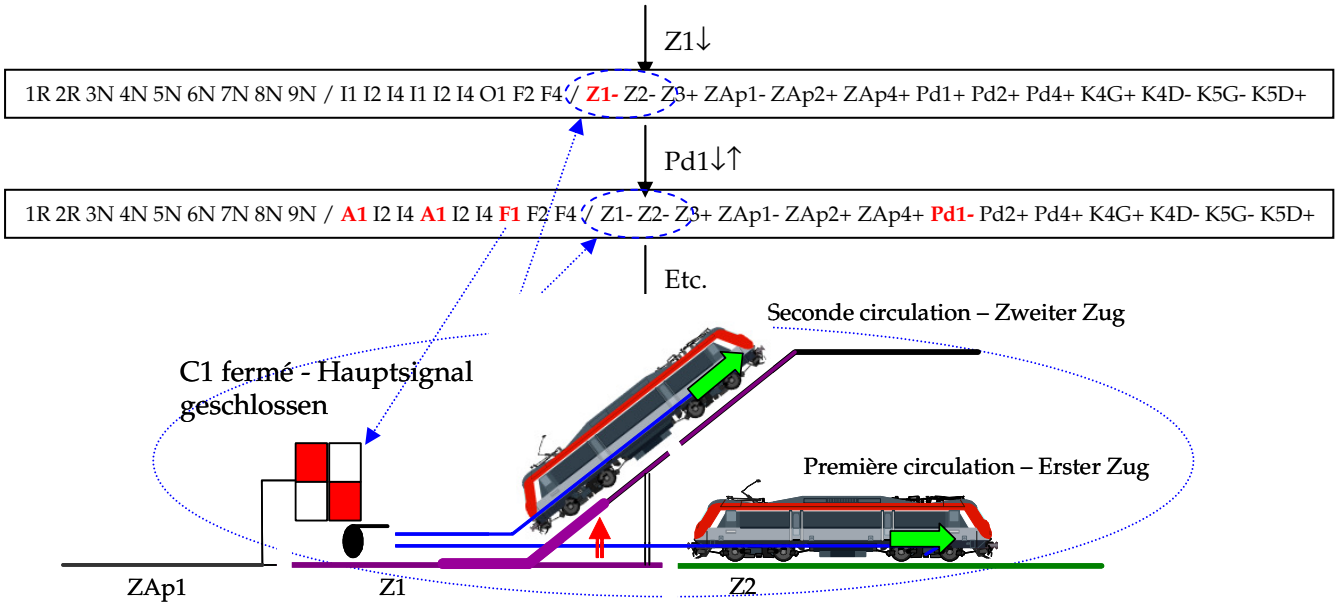



**Figure 7.21** : Libération au plus tôt de l'aiguille correspondants au passage successif de deux circulations
**Abbildung 7.21**: Weiche frei gefahren, was dem Vorbeifahren von zwei sich folgenden Zügen entspricht

**La connaissance exhaustive des états système accessibles constitue la première étape de notre démarche de validation formelle**. Le cas échéant, le domaine des états système peut être limité en éliminant les combinaisons interdites des entrées (impossibilités physiques ou règlementaires).

L'étape suivante nécessite de formaliser les propriétés de sécurité à prouver. Elles doivent être complètes pour garantir la sécurité des circulations. Celles-ci doivent être formalisées formellement, indépendamment de la réalisation du fonctionnel. C'est ce que nous allons aborder au paragraphe 7.2.2.4.

A titre d'exemple, la figure suivante illustre une partie de l'arbre des états accessibles obtenu automatiquement grâce à notre outil d'exploration.

**Die komplette Kenntnis der zugänglichen Systemzustände ist die erste Etappe der formalen Validierungsmethode**. Der Bereich der Systemzustände kann durch die Eliminierung verbotener Eingangskombinationen eingegrenzt werden (physikalisch oder vorschriftsmäßig unmöglich).

Die folgende Etappe erfordert, die zu beweisenden Sicherheitseigenschaften zu formalisieren. Um die Sicherheit der Züge zu garantieren müssen diese vollständig sein. Die Sicherheitseigenschaften müssen unabhängig von der Verwirklichung der Funktionen formalisiert werden. Dies wird im Abschnitt 7.2.2.4 erläutet.

Als Beispiel illustriert Abb. 7.22 einen Teil des Baumes der erreichbaren Zustände, der automatisch mit dem hier vorgestellten Auswertungsprogramm erhalten wurde.

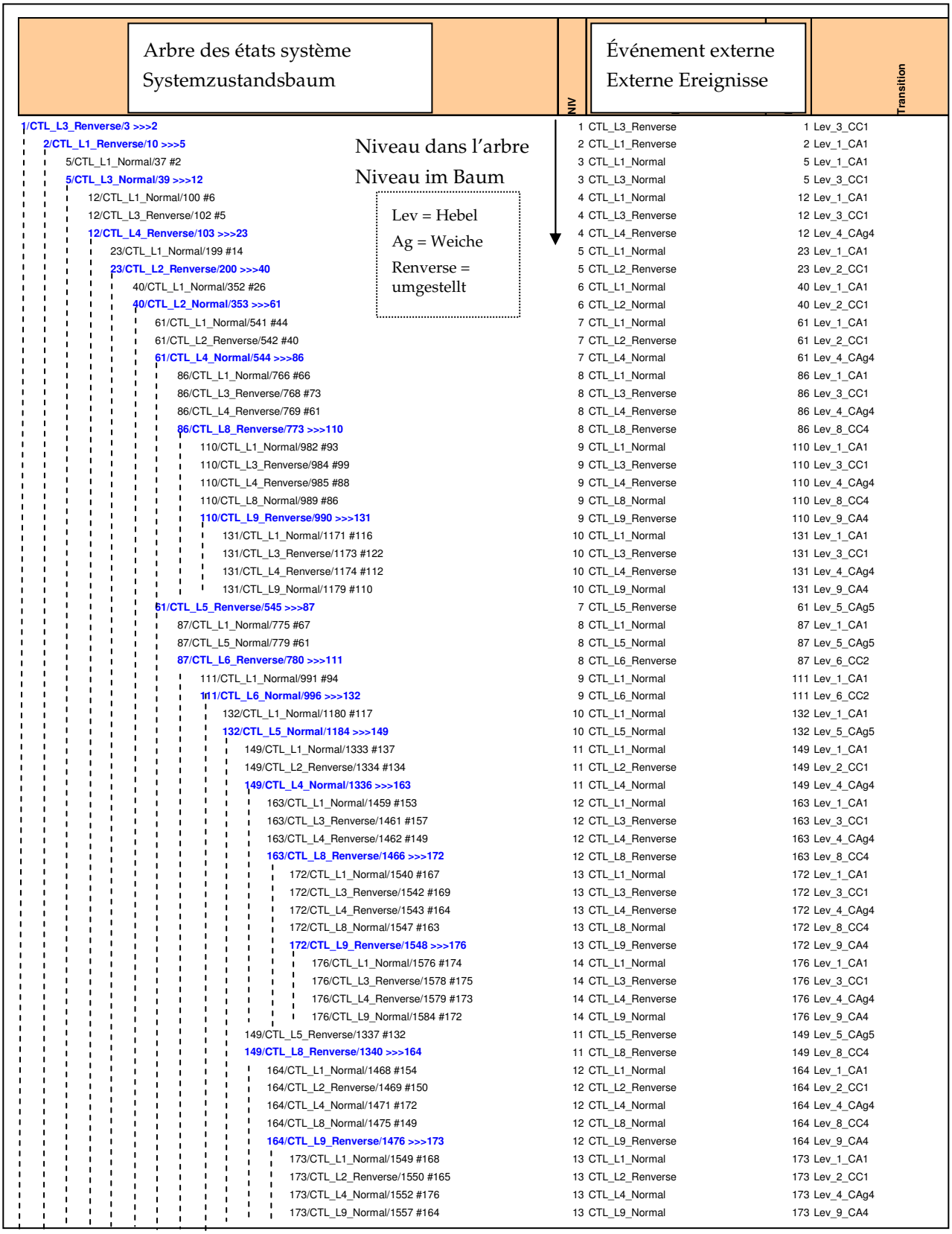



**Figure 7.22** : Extrait de l'arbre des états accessibles depuis l'état initial du poste d'aiguillage (en bleu les événements produisant un état système nouveau – en noir les événements produisant un état système identique à un état connu)
**Abbildung 7.22**: Teil des Baumdiagramms der erreichbaren Zustände vom Anfangszustand des Stellwerks ausgehend (in blau die Ereignisse, die einen neuen Systemzustand erzeugen, in schwarz die Ereignisse, die einen zu einem bekannten Systemzustand identischen Zustand erzeugen).

La figure 7.22 présente les 60 premières feuilles de l'arbre des états système sur un total de 323 états système.

Abbildung 7.22 zeigt die ersten 60 Blätter des Systemzustandsbaums, der insgesamt 323 Zustände hat.

Le nombre d'états système identifiés par l'exploration automatique est fonction du périmètre des postulats retenus pour l'exploration.

Ainsi, suivant le périmètre à explorer, nous obtenons les résultats du tableau 7.8.

Situation A :

Tous les itinéraires du poste (4 itinéraires) :

- Situation A1: Sans aucun enclenchement électrique ;
- Situation A2 : Existence des entrées et enclenchements électriques suivants : ZAp, Aig, KAg ;
- Situation A3 : Existence de tous les entrées et enclenchements électriques.

Situation B :

Tous les itinéraires du poste sauf l'itinéraire C vers A (C4 restant fermé) :

- - Situation B1: Sans enclenchements électriques
- - Situation B3 : Existence de toutes les entrées et enclenchements électriques.

| | Nbre états nouv. | Nbre transitions testées | Transition vrai | Transition P5 |
|---|---|---|---|---|
| A1 | 51 | 1452 | 170 | 0 |
| A2 | 159 | 6104 | 552 | 0 |
| A3 | 323 | 15888 | 1265 | 0 |
| B1 | 37 | 850 | 110 | 0 |
| B3 | 77 | 3374 | 244 | 0 |

**Tableau 7.8** : Synthèse des résultats obtenus avec l'outil de preuve

Logiquement, le nombre d'état est plus élevé quand les entrées électriques sont activées (en l'occurrence 323 par rapport à 51). De même, la réduction des possibilités du poste de 4 à 3 itinéraires possibles induit une réduction du nombre d'états systèmes accessibles de 323 à 77 états systèmes distincts.

Le nombre d'états accessibles est fini, dénombrable en moins 2 minutes sur un PC bureautique.

Die Anzahl der Systemzustände, die durch die automatische Auswertung identifiziert werden, hängt vom Umfang der Anforderungen an die Auswertung ab.

Je nach, auszuwertendem Bereich, erhält man die Ergebnisse der Tabelle 7.8.

Situation A:

Alle Fahrstraßen des Stellwerkes (vier Fahrstraßen):

- Situation A1: ohne elektrische Verschlüsse
- Situation A2: Existenz von Eingängen und von elektrischen Verschlüssen: ZAp, Aig und KAg
- Situation A3: Existenz von allen Eingängen von elektrischen Verschlüssen.

Situation B:

Alle Fahrstraßen des Stellwerkes außer der Strecke von C nach A (C4 bleibt geschlossen):

- Situation B1: ohne elektrische Verschlüsse
- Situation B3: Existenz aller Eingänge und aller elektrischen Verschlüsse.

| | Neue Zustandsanzahl | Anzahl der getesteten Übergänge | Wahre Übergänge | Übergang P5 |
|---|---|---|---|---|
| A1 | 127 | 1152 | 392 | 0 |
| A2 | 159 | 6104 | 552 | 0 |
| A3 | 323 | 15888 | 1265 | 0 |
| B1 | 37 | 850 | 110 | 0 |
| B3 | 77 | 3374 | 244 | 0 |

**Tabelle 7.8**: Zusammenfassung der mit dem Auswertungsprogramm erhaltenen Ergebnisse

Es ist logisch, dass die Zustandsanzahl höher ist, wenn die elektrischen Eingänge aktiviert sind (in diesem Fall 323 in Vergleich zu 51 Systemzuständen). Genauso bewirkt die Reduzierung der Kapazität des Stellwerks von 4 auf 3 mögliche Fahrstraßen eine Reduzierung der erreichbaren Systemzustände von 323 auf 77 unterschiedliche Zustände.

Die Anzahl der erreichbaren Systemzustände ist endlich und in weniger als zwei Minuten mit einem normalen PC auflistbar.

### 7.2.2.4 Propriétés de sécurité

Il s'agit maintenant d'identifier clairement les états système qui ne doivent jamais être atteints pour garantir la sécurité des circulations, au sens des agents chargés de la validation du système.

Pour ce faire nous allons définir pour chaque itinéraire du poste les propriétés de sécurité devant être toujours respectée. Ces propriétés évoluant au gré de la situation fonctionnelle du poste, ces propriétés sont écrites sous forme d'automates de preuve (AP) décrivant les enclenchements nécessaires, indépendamment du modèle fonctionnel décrit précédemment.

A titre d'exemple, la figure 7.23 illustre les automates de preuves utiles à l'itinéraire A vers C.

### 7.2.2.4 Sicherheitseigenschaften

Es handelt sich jetzt darum, die Systemzustände die aufgrund der Gewährleistung der Sicherheit der Züge niemals erreicht werden dürfen (im Sinne der Signaltechnikingenieure, die das System validieren) klar zu identifizieren.

Dazu werden für jede Fahrstraße des Stellwerks die Sicherheitseigenschaften definiert, die zu jedem Zeitpunkt eingehalten werden müssen. Diese Eigenschaften verändern sich mit dem funktionellen Zustand des Stellwerkes. Sie werden in Form von Beweisautomaten formuliert, die die notwendigen Verschlüsse, unabhängig vom vorher beschriebenen funktionellen Modell, beschreiben.

Als Beispiel illustriert die Abbildung 7.23 die notwendigen Beweisautomaten für die Fahrstraße von A nach C.

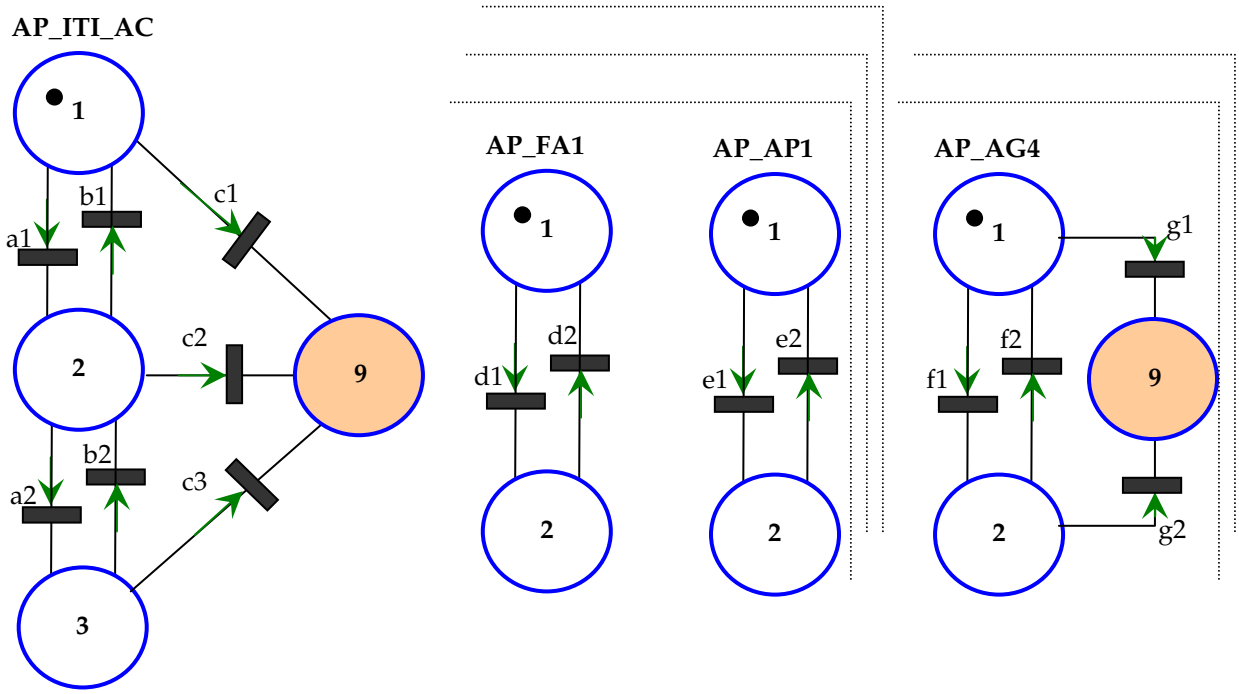



**Figure 7.23** : Automates de preuve définis pour valider le modèle du poste d'aiguillage
**Abbildung 7.23**: Beweisautomaten, die definiert werden, um das Stellwerksmodell zu validieren

L'occupation de la place 9 traduit le non respect (fugitif ou permanent) d'une ou plusieurs propriétés de sécurité.

Si, l'exploration achevée, aucune des places 9 n'est occupée, c'est donc qu'il n'existe de transition accessible par le système qui puisse ne pas respecter l'une des propriétés de sécurité.

Nous venons de montrer que les propriétés de sécurité sont toutes vérifiées quelque soit l'état courant du système (à partir de l'état initial fixé).

Le tableau 7.9 suivant décrit les transitions des automates de preuve.

Das Besetzen des Platzes 9 stellt das Nichteinhalten (flüchtig oder permanent) von einer oder mehreren Sicherheitseigenschaften dar.

Wenn nach Beenden des Beweises keiner der Stellen 9 besetzt ist, so existiert keine Transition die die Sicherheitseigenschaften nicht respektiert.

Wie gerade eben gezeigt, werden alle Sicherheitseigenschaften überprüft, unabhängig vom laufenden Zustand des Systems (von einem festen Anfangszustand ausgehend).

Tabelle 7.9 beschreibt die Transitionen der Beweisautomaten.

| | Événement / Ereignis | Condition / Bedingung | Action / Handlung |
|---|---|---|---|
| a1 | 3R | [4N et 5N et 6N et 7N]<br>[4N und 5N und 6N und 7N] | / |
| a2 | IND_CC1_O | [CTL_KA4D+ et AP_FA1_1]<br>[CTL_KA4D+ und AP_FA1_1] | / |
| b1 | 3N | [1R et (CTL_ZAp1+ ou AP_AP1_2)]<br>[1R und (CTL_ZAp1+ oder AP_AP1_2)] | / |
| b2 | CTL_KA4D- ou AP_FA1_1<br>CTL_KA4D- oder AP_FA1_1 | [IND_CC1_F] | / |
| c1 | 3R | [non (4N et 5N et 6N et 7N)]<br>[nicht (4N und 5N und 6N und 7N)] | /P_5; |
| c2 | 3N | [non (1R et (CTL_ZAp1+ ou AP_AP1_2))]<br>[nicht (1R und (CTL_ZAp1+ oder AP_AP1_2))] | /P_5; |
| c2 | 1R ou 4R ou 5R ou 6R ou 7R<br>1R oder 4R oder 5R oder 6R oder 7R | [3R] | /P_5; |
| c2 | IND_CC1_O | [non (CTL_KA4D+ et AP_FA1_1)]<br>[nicht (CTL_KA4D+ und AP_FA1_1)] | /P_5; |
| c3 | CTL_KA4D- ou AP_FA1_1<br>CTL_KA4D- oder AP_FA1_1 | [non (IND_CC1_F)]<br>[nicht (IND_CC1_F)] | /P_5; |
| d1 | CTL_Pd1- ou CTL_Z1-<br>CTL_Pd1- oder CTL_Z1- | [CTL_Pd1- et CTL_Z1- et (2R ou 3R)]<br>[CTL_Pd1- und CTL_Z1- und (2R oder 3R)] | /AP_FA1_2; |
| d2 | 2N ou 3N<br>2N oder 3N | [2N et 3N]<br>[2N und 3N] | /AP_FA1_2; |
| e1 | CTL_Pd1- ou CTL_Z1-<br>CTL_Pd1- oder CTL_Z1- | [CTL_ZAp1- et CTL_Z1- et AP_FA1_2 et CTL_Pd1- et (2R ou 3R)]<br>[CTL_ZAp1- und CTL_Z1- und AP_FA1_2 und CTL_Pd1- und (2R oder 3R)] | /AP_AP1_2; |
| e2 | 2N ou 3N<br>2N oder 3N | [2N et 3N]<br>[2N und 3N] | /AP_AP1_1; |
| f1 | 4R | [1N et 2N et 3N et CTL_Z1+]<br>[1N und 2N und 3N und CTL_Z1+] | / |
| f2 | 4N | [1N et 2N et 3N et 6N et 7N et CTL_Z1+]<br>[1N und 2N und 3N und 6N und 7N und CTL_Z1+] | / |
| g1 | 4R | [(1R ou 2R ou 3R) ou CTL_Z1-]<br>[(1R oder 2R oder 3R) oder CTL_Z1-] | /P_5; |
| g2 | 4N | [(1R ou 2R ou 3R ou 6R ou 7R) ou CTL_Z1]<br>[(1R oder 2R oder 3R oder 6R oder 7R) oder CTL_Z1] | /P_5; |

**Tableau 7.9** : description des automates de preuve

**Tabelle 7.9**: Beschreibung der Beweisautomaten

### 7.2.2.5 Arbre des états systèmes accessibles et prouvés du modèle

Le vecteur d'état du poste est complété par le vecteur d'état des automates de preuve VE' :

VE' = [IT_AB IT_AC IT_BA IT_CA FA1 FA2 FA4 AP1 AP2 AP4 AG4 AG5] **(21)**

Le vecteur d'état système VS est obtenu comme la concaténation des vecteurs VE et VE' :

VS = [VE VE'] **(22)**

Cette opération ne doit pas augmenter le nombre d'états système par rapport à celui obtenu à l'exploration sans le complément VE'. Les automates de preuves (AP) doivent traduire la vision des propriétés attendues du système indépendamment de sa réalisation, mais fonctionnellement recouvrent les mêmes réalités.

### 7.2.2.5 Systemszustandbaum des Models

Der Zustandsvektor des Stellwerks wird durch den Zustandsvektor der Beweisautomaten VE' ergänzt:

VE' = [IT_AB IT_AC IT_BA IT_CA FA1 FA2 FA4 AP1 AP2 AP4 AG4 AG5] **(21)**

Den Systemzustandsvektor VS erhält man durch die Verkettung der Vektoren VE und VE':

VS = [VE VE'] **(22)**

Diese Verkettung darf nicht die Anzahl der Systemzustände im Vergleich zur Anzahl, die man durch das Auswerten ohne den Zusatz VE' erhalten hätte, erhöhen. In der Tat sollen die Beweisautomaten die Vorstellung der erwarteten Eigenschaften unabhängig von der Umsetzung darstellen, jedoch funktionell die gleiche Wirklichkeit abdecken.

Un accroissement du nombre d'états système traduit alors une divergence entre les visions «conception – modélisation» et «sécurité».

La présentation sous forme d'arbre des états systèmes accessibles résultant de l'exploration et de la réalisation de la preuve pour chaque transition valide (vrai) est trop volumineuse pour être présentée de manière complète. Nous allons présenter les résultats sous une forme simplifiés dite « arbres des événements accessibles».

L'automatisation du traitement appliqué à notre modèle de poste mécanique ne révèle aucune transition d'automate de preuve vers une place 9. Il n'existe donc pas de combinaisons et/ou séquences autorisées des entrées (manœuvre de levier, changement d'état des entrées électriques) qui s'affranchissent au moins d'une propriété de sécurité.

***NB** : Notons que les séquences non autorisées, non admissibles compte tenu des règles d'exploitation du poste (refoulement de train par exemple) ont été interdites à l'exploration.*

Eine Zunahme der Systemzustände deckt ein Auseinanderklaffen der Vorstellungen „Konzeption – Beschreibung" und „Sicherheit" auf.

Die Darstellung der erreichbaren Systemzustände in Form eines Baumdiagramms, das durch das Auswerten und die Durchführung des Beweises für jede validierte (wahre) Transition erhalten wird, ist zu umfangreich um hier vollständig dargestellt zu werden. Deshalb werden die Ergebnisse in einer vereinfachten Form dargestellt, den so genannten „Bäumen der erreichbaren Ereignisse". Die Automatisierung der auf das Modell des mechanischen Stellwerks angewandten Bearbeitung weist keine Transition des Beweisautomaten zu einem Platz 9 auf. Es gibt also keine Kombination und/oder Eingangssequenz (Umstellen des Hebels, Änderung des Zustands der Eingänge) die mindestens eine Sicherheitseigenschaft verletzt.

***NB**: Es ist anzumerken, dass die nicht erlaubten, aufgrund der Betriebsregeln des Stellwerks nicht zugelassenen Sequenzen bei der Analyse verboten sind.*

Pour illustration voici le début de l'arbre des événements pour le cas B1 sans erreur

Zur Illustration zeigt Abb. 7.24 den Anfang des Ereignisbaums für den Fall B1 ohne Fehler

*Renversé = umgestellt*

| | |
|---|---|
| **1/CTL_L3_Renverse/3 >>>2** | 1 CTL_L3_Renverse |
| **2/CTL_L1_Renverse/8 >>>4** | 2 CTL_L1_Renverse |
| 4/CTL_L1_Normal/22 #2 | 3 CTL_L1_Normal |
| **2/CTL_L3_Normal/10 >>>5** | 2 CTL_L3_Normal |
| 5/CTL_L3_Renverse/31 #2 | 3 CTL_L3_Renverse |
| **5/CTL_L4_Renverse/32 >>>8** | 3 CTL_L4_Renverse |
| **8/CTL_L2_Renverse/51 >>>13** | 4 CTL_L2_Renverse |
| **13/CTL_L1_Renverse/85 >>>21** | 5 CTL_L1_Renverse |
| 21/CTL_L1_Normal/141 #13 | 6 CTL_L1_Normal |
| **21/CTL_L5_Renverse/145 >>>31** | 6 CTL_L5_Renverse |
| 31/CTL_L1_Normal/211 #23 | 7 CTL_L1_Normal |
| 31/CTL_L5_Normal/215 #21 | 7 CTL_L5_Normal |
| **31/CTL_L6_Renverse/216 >>>44** | 7 CTL_L6_Renverse |
| 44/CTL_L1_Normal/302 #34 | 8 CTL_L1_Normal |
| **44/CTL_L6_Normal/307 >>>52** | 8 CTL_L6_Normal |
| 52/CTL_L1_Normal/358 #46 | 9 CTL_L1_Normal |
| **52/CTL_L5_Normal/362 >>>60** | 9 CTL_L5_Normal |
| 60/CTL_L1_Normal/414 #56 | 10 CTL_L1_Normal |
| 60/CTL_L5_Renverse/418 #52 | 10 CTL_L5_Renverse |
| 52/CTL_L6_Renverse/363 #44 | 9 CTL_L6_Renverse |
| **44/CTL_L7_Renverse/308 >>>53** | 8 CTL_L7_Renverse |
| 53/CTL_L1_Normal/365 #47 | 9 CTL_L1_Normal |
| 53/CTL_L7_Normal/371 #44 | 9 CTL_L7_Normal |
| **13/CTL_L2_Normal/86 >>>22** | 5 CTL_L2_Normal |
| 22/CTL_L2_Renverse/149 #13 | 6 CTL_L2_Renverse |
| **22/CTL_L4_Normal/151 >>>32** | 6 CTL_L4_Normal |
| 32/CTL_L3_Renverse/220 #26 | 7 CTL_L3_Renverse |
| 32/CTL_L4_Renverse/221 #22 | 7 CTL_L4_Renverse |

**Figure 7.24** : Extrait de l'arbre des transitions franchissables de la situation B1 depuis l'état initial du poste d'aiguillage
**Abbildung 7.24**: Auszug aus dem Baumdiagramm der schaltbaren Transitionen der Situation B1 vom Anfangzustand des Stellwerks ausgehend

Pour illustration voici le début de l'arbre pour le cas B1 avec erreur

Un enclenchement a été volontairement retiré du fonctionnel, en l'occurrence L4N/L2N.

Zur Illustration zeigt Abb. 7.25 den Anfang des Ereignisbaums für den Fall B1 mit Fehler. Der Verschluss L4N/L2N ist aus den Funktionen herausgenommen worden.

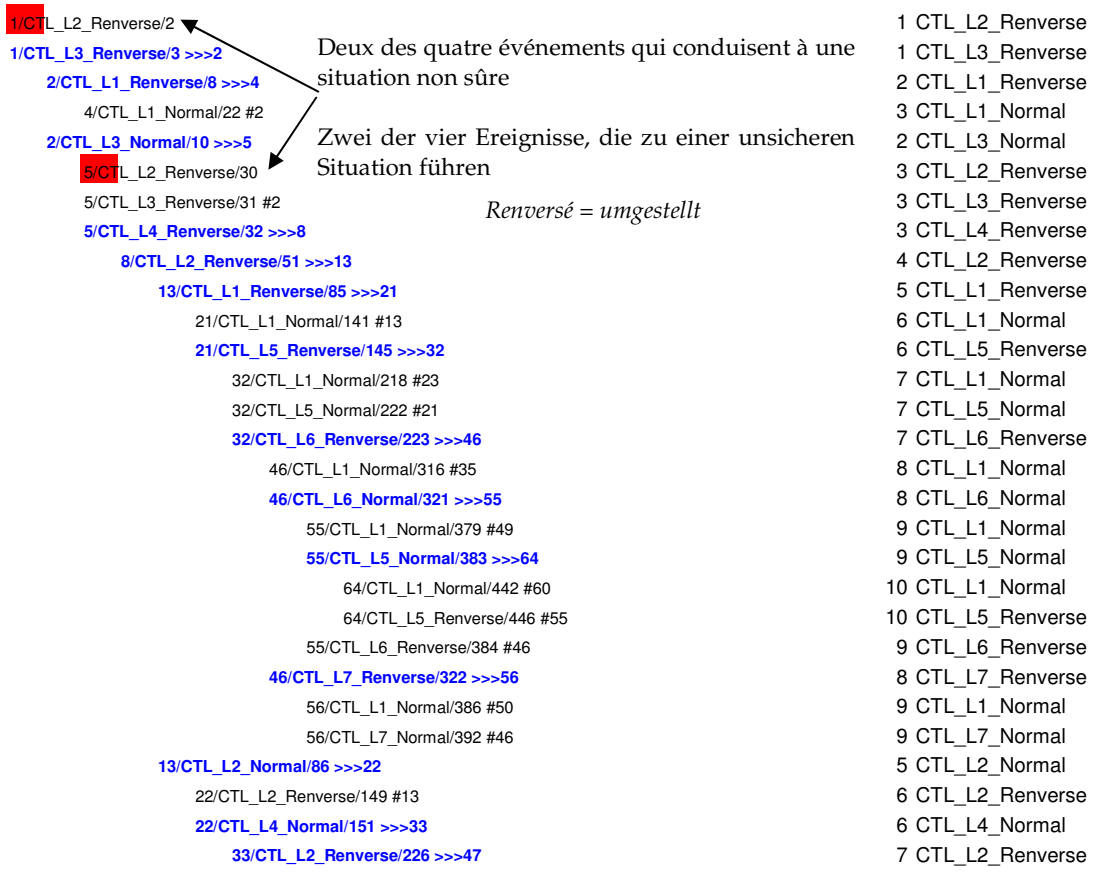



**Figure 7.25** : Extrait de l'arbre de la situation B1 avec l'absence de l'enclenchement L4N/L2N
**Abbildung 7.25**: Abschnitt des Baumes für die Situation B1 mit der Abwesenheit des Verschlusses L4N/L2N

Notons que l'introduction d'une absence d'un enclenchement (ce serait de même pour une incompatibilité complète) conduit à de multiples déclenchements, rejetant la preuve tentée. Cette même erreur conduit dans les situations précédentes aux résultats suivants :

| | Nbre États Nouv. | Nombre de transitions | Transition Vrai | Transition P_5 |
|---|---|---|---|---|
| A3 | 151 | 1368 | 472 | 12 |

**Tableau 7.10 :** Synthèse des résultats obtenus avec l'outil de preuve (B1 avec erreur)

La dernière colonne traduit le nombre de contre exemple existant qui ne respecte pas a minima une propriété. Dans notre cas, il peut y avoir de nombreux contre exemples selon la situation. Cela traduit l'existence d'arrangements d'événements externes ayant la même conséquence.

Die Abwesenheit eines Verschlusses (beim Fehlen einer vollständigen Unvereinbarkeit wäre es dasselbe) führt zu verschiedenen Auslösern und verwirft den Beweisversuch. Dieser Fehler führt in der vorangehenden Situation zu den Ergebnissen der Tabelle 7.10.

| | Neue Zustandsanzahl | Anzahl der getesteten Übergänge | Wahre Übergänge | Übergang P5 |
|---|---|---|---|---|
| A3 | 151 | 1368 | 472 | 12 |

**Tabelle 7.10:** Zusammenfassung der mit dem Auswertungsprogramm erhaltenen Ergebnisse (B1 mit Fehler)

Die letzte Spalte gibt die Anzahl bestehender Gegenbeispiele an, die mindestens eine Eigenschaft nicht respektieren. In obigem Fall, kann es je nach Situation mehrere Gegenbeispiele geben. Das zeigt die Existenz von Anordnungen externer Ereignisse, die dieselbe Folge haben.

Les outils développés permettent de faciliter l'analyse de ces contre-exemples en permettant :

- de retrouver les transitions menant à des états système non surs ;
- de visualiser la chronologie des événements externes menant à ces transitions non sûres ;
- de visualiser les vecteurs d'état à chaque événement externe ;
- de visualiser les transitions des automates de preuve (AP) qui sont mis en cause : propriété de sécurité non vérifiée.

Illustrons ces possibilités d'analyse sur des écrans apparaissant sur la situation A1 (Figure 7.26) :

Die entwickelten Werkzeuge ermöglichen es, die Analyse dieser Gegenbeispiele zu vereinfachen und erlauben es:

- die Transitionen wiederzufinden, die zu den unsicheren Systemzuständen geführt haben.
- die Chronologie der zu diesen unsicheren Transitionen führenden externen Ereignisse sichtbar zu machen.
- die Zustandsvektoren zum Zeitpunkt jedes externen Ereignisses sichtbar zu machen.
- die betroffenen Transitionen der Beweisautomaten (AP) sichtbar zu machen: nicht überprüfte Sicherheitseigenschaft.

Die Untersuchungsmöglichkeiten des Analysewerkzeugs werden anhand der Situation A1 in Abbildung 7.26 dargestellt.

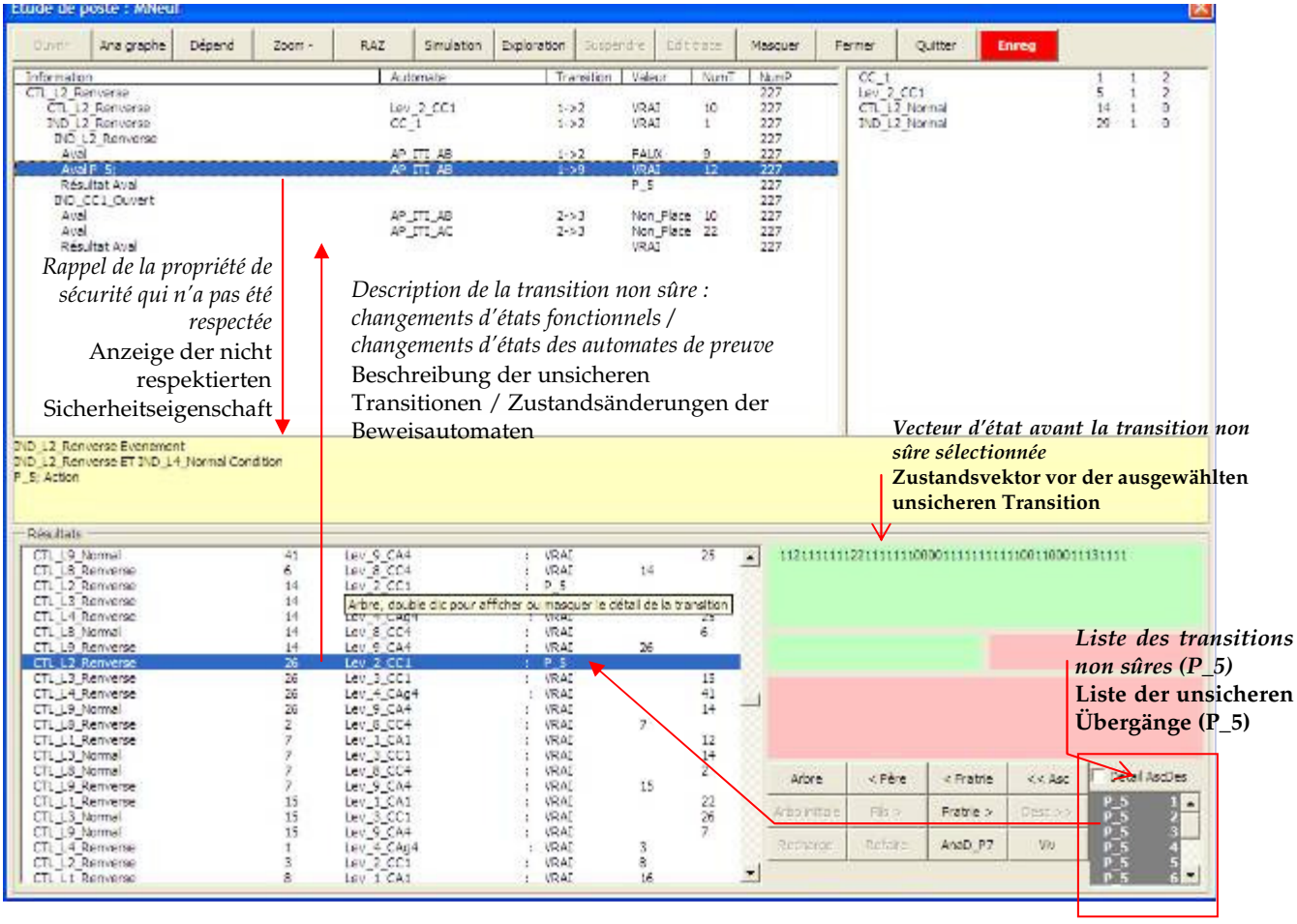



**Figure 7.26** : Outil d'analyse des résultats dans le cadre d'un échec de la preuve – existence de contre exemples-situation A1 avec l'absence de l'enclenchement L4N/L2N

**Abbildung 7.26**: Analysewerkzeug der Ergebnisse bei einem Misserfolg des Beweises – Existenz von Gegenbeispielen - Situation A1 mit der Abwesenheit des Verschlusses L4N/L2N

Les outils développés permettent d'analyser les contre exemples afin d'identifier si l'erreur revient à la conception et/ou à l'écriture des propriétés de sécurité. La preuve devra être relancée une fois les correctifs réalisés sur le fonctionnel et/ou sur les automates de preuve.

Il est remarquable de noter que l'application de notre méthode à une table d'enclenchement mécanique (indépendamment des automatismes électriques ajoutés ultérieurement - A1) est équivalente à l'application de la méthode « Cossmann Descubes » [Descubes, 1898]. Cette méthode est utilisée pour la validation (formelle avant l'heure) des tables mécaniques d'enclenchement des postes d'aiguillage en France.

Partant des incompatibilités retenues pour un poste, elle montre notamment s'il existe des enclenchements surabondants (possibilité de blocage de la table), s'il existe des enclenchements manquants. Enfin, elle donne la liste des mouvements de levier à opérer (arbre des mouvements à compter de l'état initial de la table) pour vérifier exhaustivement la table. Une exploration manuelle en quelque sorte.

Il y a des analogies surprenantes entre ces deux approches. Tous les postes mécaniques ou électromécaniques étaient/sont ainsi validés exhaustivement après toute intervention (modification ou révision). Il en est de même pour les postes électriques. Ce n'est plus le cas pour les postes informatiques… ce qui n'est pas un progrès.

**De telles démarches sont appliquées depuis le début du siècle sur les postes mécaniques. Une méthode formelle avant l'heure, il y a plus de cent ans en quelque sorte [Descubes, 1898] [SNCF, 1963].**

Die entwickelten Werkzeuge ermöglichen es, die Gegenbeispiele zu analysieren um zu identifizieren, ob der Fehler in der Konzeption und/oder im Formulieren der Sicherheitseigenschaften liegt. Der formale Beweis muss dann nach der Korrektur der Funktionen und/oder der Beweisautomaten erneut gestartet werden.

Es ist bemerkenswert festzustellen, dass die Anwendung der hier vorgestellten Methode auf ein mechanisches Verschlussregister (unabhängig von den später hinzugefügten elektrischen Steuerungen - A1) gleichwertig ist mit der Anwendung der „Cossmann Descubes"-Methode [Descubes, 1898]. Diese Methode (damals schon formal) wird in Frankreich für die Prüfung des mechanischen Verschlussregisters der Stellwerke benutzt.

Von den für ein Stellwerk definierten Unvereinbarkeiten ausgehend, zeigt die Methode insbesondere, ob es überflüssige Verschlüsse gibt (Möglichkeit das Register zu blockieren) und ob es fehlende Verschlüsse gibt. Schließlich erzeugt sie auch die Liste der Hebelbewegungen (Baumdiagramm der Bewegungen vom Anfangszustand des Tisches aus) die nötig sind, um das Register manuell vollständig zu überprüfen; - gewissermaßen eine manuelle Auswertung.
Es gibt überraschende Analogien zwischen diesen zwei Methoden. Alle mechanischen oder elektromechanischen Stellwerke wurden/werden so nach jedem Eingriff geprüft (Änderung oder Inspektion). Das gleiche gilt für die elektrischen Stellwerke. Dies gilt jedoch nicht mehr bei IT-Stellwerken… was kein Fortschritt ist.

**Vorgehensweisen, die der hier vorgestellten Methode ähneln werden seit Anfang letzten Jahrhunderts bei mechanischen Stellwerken angewandt: eine Methode, die unbewusst schon formal war, vor mehr als einhundert Jahren [Descubes, 1898] [SNCF, 1963].**

## 7.3 Poste PIPC de Nurieux

La méthode a été appliquée sur le fonctionnel de plusieurs postes réels de type PIPC (en service ou devant l'être prochainement) au moyen d'outils automatisant son exécution. Les graphes fonctionnels (qui seront réellement interprétés par les machines cibles) ont fait l'objet du travail.

Présenterons les résultats obtenus sans entrer dans un niveau de détail élevé. Afin de vérifier l'efficacité de la démarche nous avons introduit volontairement des erreurs de conception des graphes de différentes natures, à différents niveaux de fonctionnement… action qui a toujours donné lieu à une détection justifiée par les outils, validant ainsi d'une part les outils, d'autre part la méthode dans son application concrète sur des cas industriels.

### 7.3.1 Présentation du plan de voie et du programme du poste

Le poste de Nurieux est un petit poste (16 itinéraires, 1 autorisation, 1 passage à niveau) de ligne de voie unique avec block automatique. Ce poste reprend néanmoins environ 60% des principes génériques des postes actuellement en exploitation en France.

La figure 7.27 présente une des zones d'action du poste telle que les agents chargés de la validation du poste avant mise en exploitation l'ont saisi. Un double clic sur chaque objet fait apparaître une fenêtre, non figurée ici, de questionnement, de paramétrage, de visualisation des paramètres identifiés topologiquement.

## 7.3 Elektronisches Stellwerk Nurieux

Die Methode wurde mit Hilfe der Werkzeuge, die ihre Durchführung automatisieren, auf die Funktionen mehrerer realer PIPC-Stellwerke (in Betrieb oder demnächst in Betrieb) angewandt. Die funktionellen Graphen (die von den Zielmaschinen direkt interpretiert werden) waren Teil dieser Arbeit.

In diesem Abschnitt werden nun einige Ergebnisse vorgestellt. Um die Wirksamkeit der Vorgehensweise zu überprüfen, wurden absichtlich Fehler in die Konzeption der Graphen eingearbeitet und dies auf verschiedenen Funktionsebenen. Diese Fehler werden immer von den Werkzeugen entdeckt. Das validiert einerseits die Werkzeuge und zeigt andererseits die Anwendbarkeit der Methode auf konkrete Industriefälle.

### 7.3.1 Darstellung des Gleisplans und des Stellwerksprogramms

Das Stellwerk von Nurieux ist klein (16 Fahrstraßen, 1 Rangiererlaubnis, 1 Bahnübergang), befindet sich auf einer eingleisigen Strecke und hat einen Selbstblock mit Lichtsignalen. Dieses Stellwerk benützt aber immerhin ungefähr 60 % der allgemeinen Prinzipien der sich in Betrieb befindlichen französischen Stellwerke.

Die Abbildung 7.27 zeigt einen Befehlsbereich des Stellwerks, wie ihn die mit der Validierung beauftragten Signaltechniker eingetragen haben. Ein Doppelklick auf ein Objekt lässt ein hier nicht dargestelltes Fenster erscheinen, mit Fragen, Parametrisierung und der Anzeige der topographisch identifizierten Parameter.

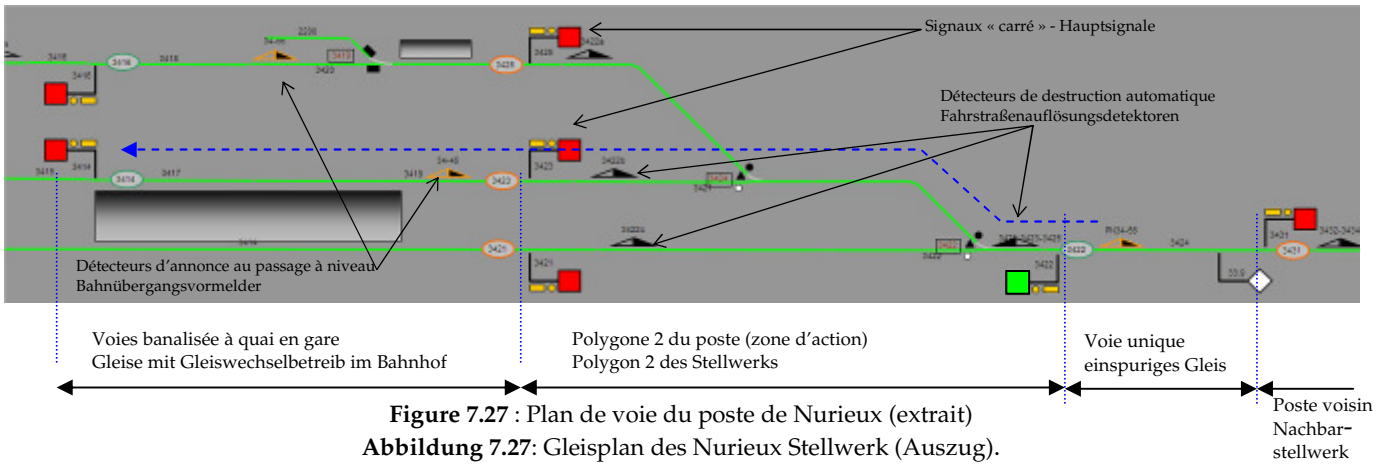



**Figure 7.27** : Plan de voie du poste de Nurieux (extrait)
**Abbildung 7.27**: Gleisplan des Nurieux Stellwerk (Auszug).

Dans la suite de l'exposé, nous allons présenter la validation de l'itinéraire 3422 vers 3414 (flèche bleue).

Im Weiteren wird die Validierung der Fahrstraße 3422 nach 3414 (blauer Pfeil) vorgestellt.

## 7.3.2 Propriétés de sécurité et postulats

Les exigences de sécurité relatives à cet itinéraire sont identifiées par les agents chargés de la validation du poste avant sa mise en exploitation finale :

- existence ou non de fonctions d'enclenchement attachées à l'itinéraire en fonction du programme fonctionnel attendu du poste ;
- paramétrage de ces fonctions en fonction de la topologie, des distances réelles d'implantation des ressources (signal, aiguille, joints isolants, détecteurs…).

Ce travail s'opère indépendamment du processus d'étude (généralement externalisé par ailleurs) et notamment sans influence des choix opérés dans le processus d'étude ! Les éléments sont rassemblés, visualisés et validés par ces agents sur la base des informations retournées par les outils (notamment le plan de voie de la figure 7.27).

Ces informations permettent d'instancier automatiquement les automates de preuve génériques requis par cet itinéraire (postulats, obligations de preuve cf. Chapitre 6), les postulats utiles à l'explorateur et au prouveur.

## 7.3.2 Sicherheitseigenschaften und Anforderungen

Die Fahrstraßensicherheitsanforderungen werden durch die mit der Validierung des Stellwerks beauftragten Prüfer identifiziert:

- Existenz oder nicht Existenz von mit einer Fahrstraße verbundenen Verschlussfunktionen in Abhängigkeit des vom Stellwerk erwarteten funktionellen Programms
- Parametrisierung dieser Funktionen entsprechend der Topologie und den wirklichen Entfernungen der Ressourcen (Signal, Weiche, Isolierstoß, Melder…).

Diese Arbeit wird unabhängig vom Analyseprozess (der im Allgemeinen ausgegliedert ist) und vor allem von der in diesem Prozess getroffenen Wahl durchgeführt. Die Elemente werden von den Prüfern auf der Basis der von den Werkzeugen ausgegebenen Informationen (besonders des Gleisplans in Abbildung 7.27) gesammelt, sichtbar gemacht und validiert. Diese Informationen ermöglichen es, die allgemeinen, von dieser Fahrstraße geforderten Beweisautomaten (Anforderungen, Beweispflicht vgl. Kapitel 6) und genauso die für das Auswerten und den Beweis nützlichen Anforderungen. automatisch zu parametrisieren.

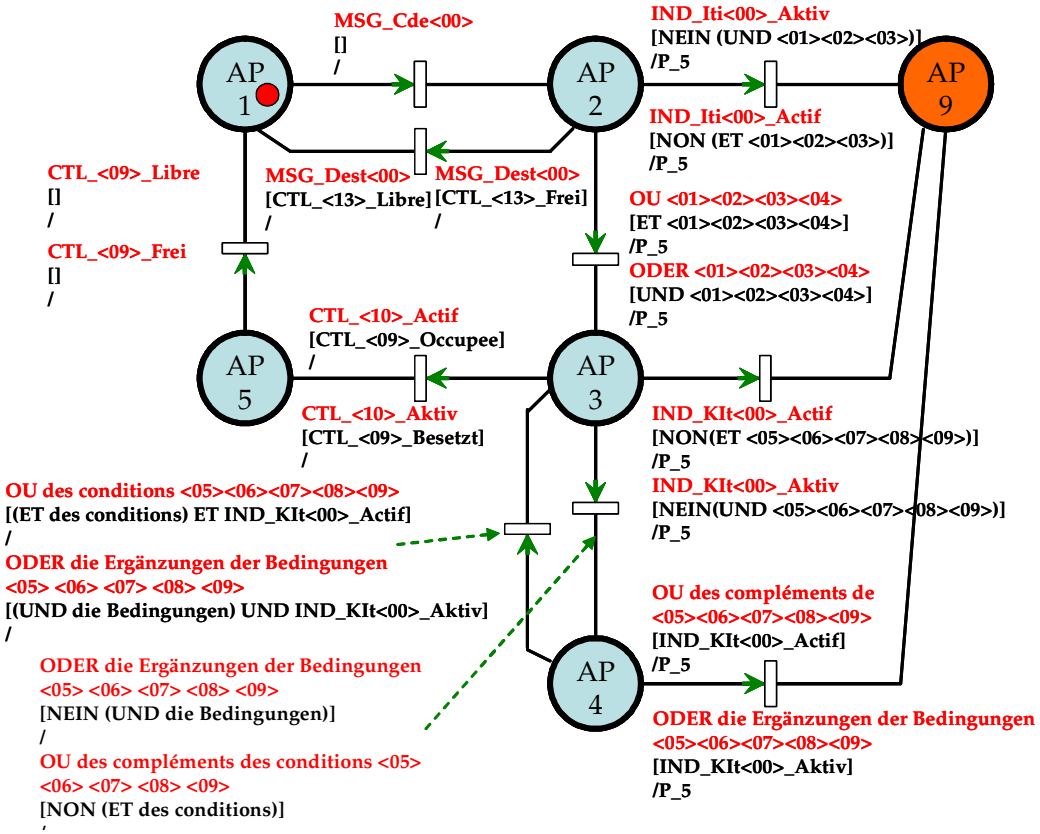



**Figure 7.28** : Automate de preuve générique (AP) pour un itinéraire
**Abbildung 7.28**: Allgemeiner Beweisautomat (AP) für eine Fahrstraße

## 7.3.3 Exploration et validation formelle du fonctionnel applicatif du poste

Les outils de mise en œuvre de notre méthode ont été utilisés pour traiter un poste d'aiguillage réel. Il s'agit d'un fonctionnel réel qui doit être mis en exploitation en fin d'année 2009, le poste de Nurieux.

Le traitement d'un itinéraire nous servira d'illustration concrète pour illustrer la procédure, les traitements et les interfaces à dispositions des utilisateurs. L'itinéraire 3322-3114 a été traité et conduit aux résultats suivants.

## 7.3.3 Auswertung und formale Validierung der Stellwerksfunktionen

Die Werkzeuge für die Anwendung der vorgestellten Methode werden in diesem Abschnitt benützt, um ein reales Stellwerk zu behandeln. Es handelt sich um ein echtes funktionelles Stellwerk, das Stellwerk von Nurieux, das Ende 2009 in Betrieb gehen soll.

Die Bearbeitung einer Fahrstraße dient als konkrete Illustration, um das Verfahren und die Bearbeitungen darzustellen. Die Fahrstraße 3322-3114 wurde behandelt und ergibt die Ergebnisse aus Abbildung 7.29.

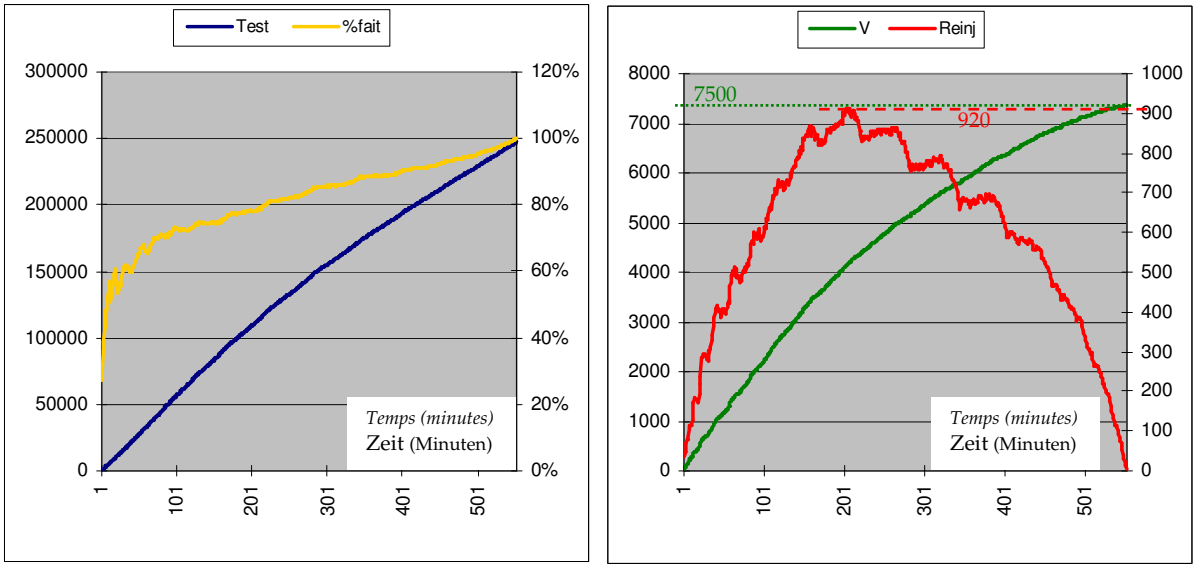



**Figure 7.29** : Résultats d'avancement des traitements automatisés d'exécution de l'exploration et de la validation formelle pour l'itinéraire 3222-3114 avec commande et destruction automatique.
**Abbildung 7.29**: Fortschritt der automatischen Bearbeitung (Auswertung und formale Validierung) für die Fahrstraße 3222-3114 mit automatischer Steuerung und Auflösung.

Le nombre de changements d'état des entrées externes explorées est de d'environ 250 000, dont:

- 25 079 d'entre elles ont été suivies de transitions valides (condition : vrai & propriétés : vrai) ;
- 117 114 ont été rejetées par le fonctionnel du poste (condition fausse) ;
- les autres transitions ont été explorées mais, ne respectant pas les postulats de fonctionnement, elles n'ont pas été réinjectés ;
- 7000 transitions fonctionnelles distinctes ont été découvertes par la méthode du pivot ;
- 920 vecteurs d'état système ont été mémorisés temporairement avant réinjections.

Die Anzahl der Zustandsänderungen der externen Eingänge beträgt ungefähr 250.000:

- 25.079 Eingängen folgten echte Transitionen (Bedingung: wahr und Eigenschaften: wahr).
- 117.114 wurden von den Funktionen des Stellwerkes abgewiesen (Bedingung: falsch).
- die anderen Übergänge wurden gefunden aber nicht eingespeist, da sie nicht den Funktionsanforderungen entsprachen.
- 7000 unterschiedliche funktionelle Übergänge wurden gefunden.
- 920 Systemzustandsvektoren wurden vorübergehend gespeichert und dann erneut eingespeist.

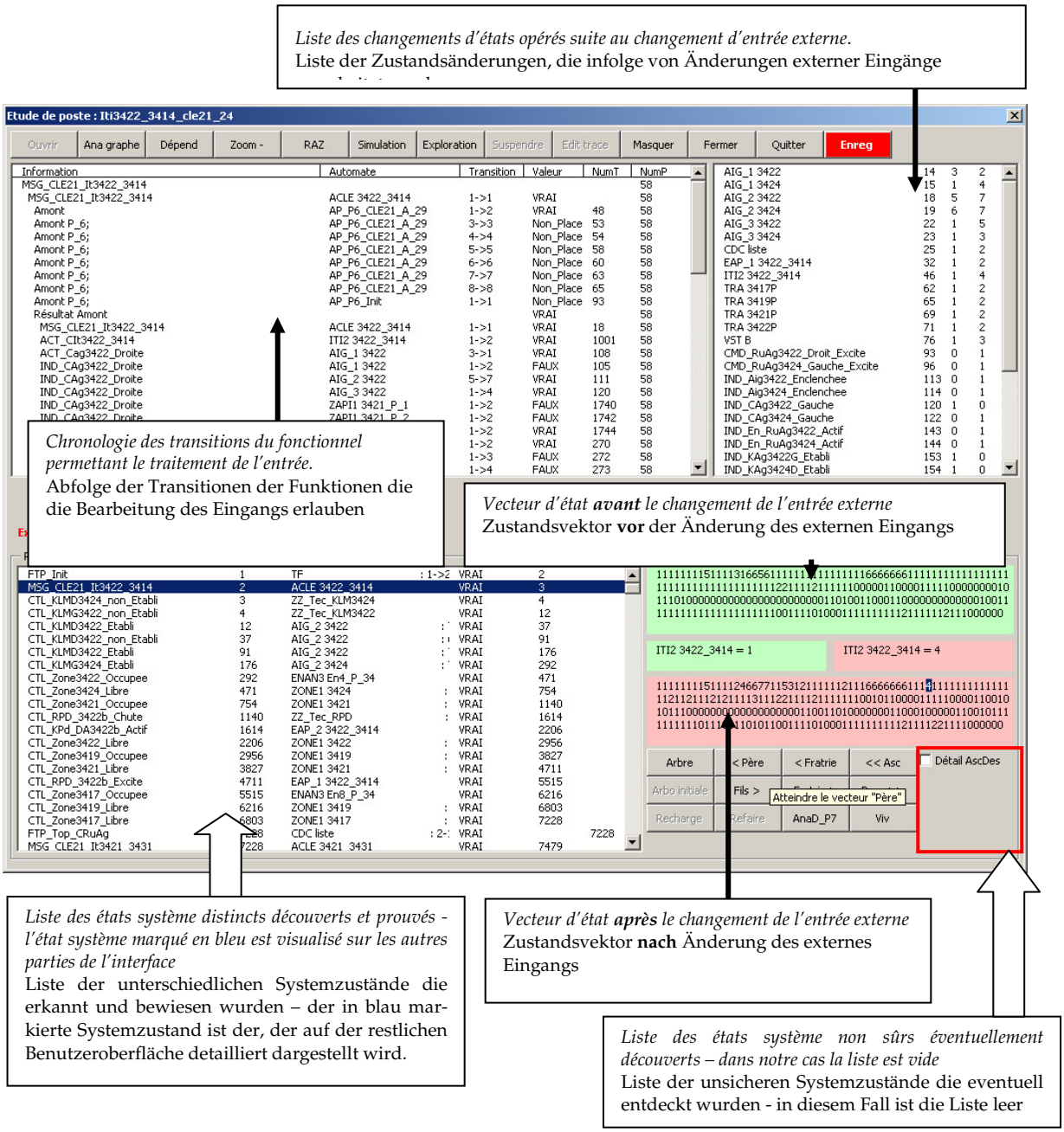



**Figure 7.30** : Tableau de contrôle des traitements automatisés pour l'itinéraire 3222-3114
**Abbildung 7.30**: Kontrolltabelle der automatisierten Verarbeitung der Fahrstraße 3222-3114

Le tableau de contrôle permet à un opérateur signalisation de suivre si besoin (fonctionnement anormal ou non respect d'une propriété de sécurité) pas à pas:

- la chronologie des événements internes du système;
- les changements d'états des graphes et des indicateurs.

L'analyse permet de détecter l'origine d'un éventuel écart entre le fonctionnement constaté et le fonctionnement attendu du point de vue des exigences de sécurité.

Die Kontrolltabelle ermöglicht es einem Prüfer falls nötig (anormale Funktionsweise oder Nichteinhaltung einer Sicherheitseigenschaft) Schritt für Schritt

- die Abfolge der internen Ereignisse des Systems
- die Änderungen der Zustände der Graphen und der Indikatoren

nachzuvollziehen.

Die Analyse ermöglicht es, den Ursprung eines eventuellen Abweichens der festgestellten Funktionsweise von den erwarteten Sicherheitsanforderungen festzustellen.

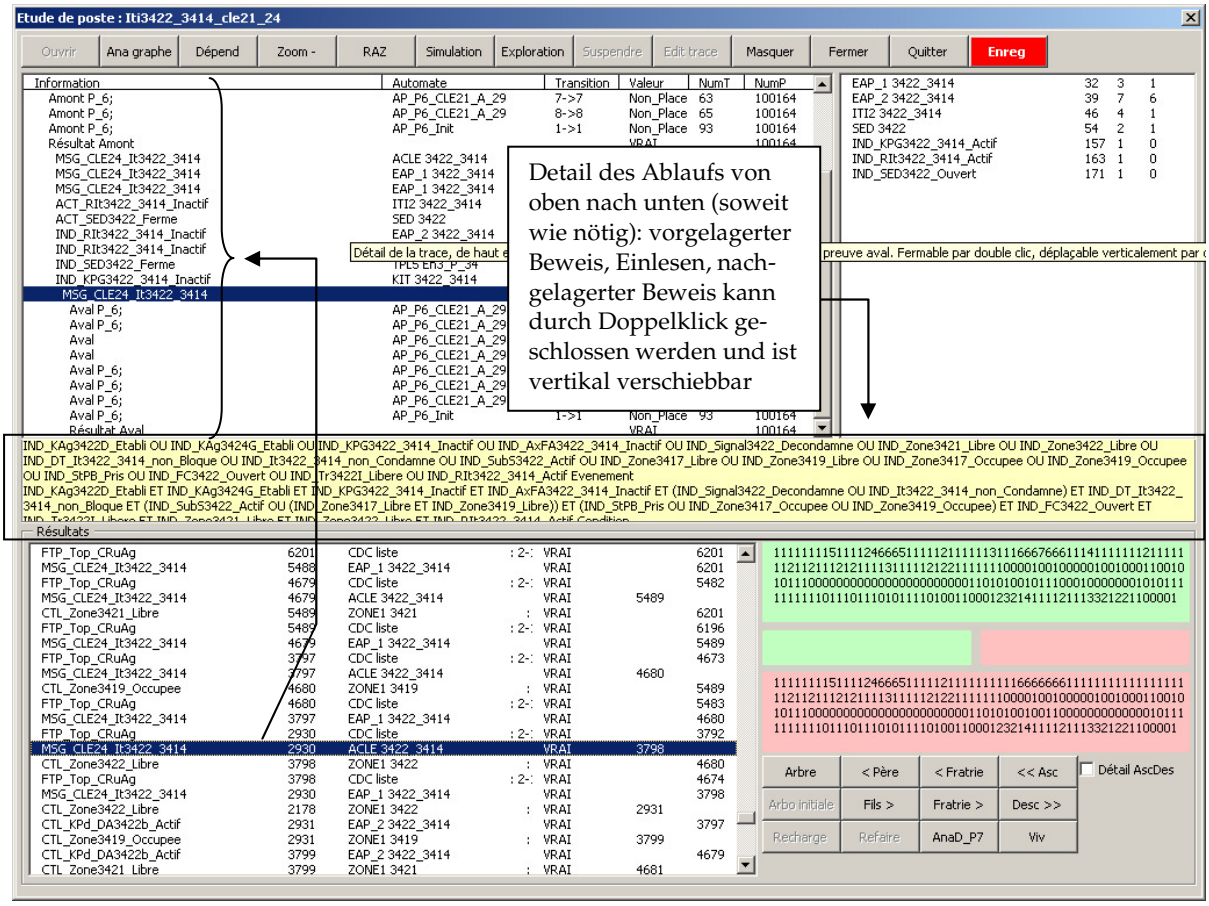



**Figure 7.31** : Tableau de contrôle des traitements automatisés pour l'itinéraire 3222-3114
**Abbildung 7.31**: Kontrolltabelle der automatisierten Verarbeitung der Fahrstraße 3222-3114

La structure fonctionnelle générique du principe de signalisation (graphes génériques) permet d'identifier les dépendances entres les itinéraires du poste. Les entrées externes qui ne participent à aucun graphe en dépendances avec ceux de l'itinéraire sont figées dans l'état de l'état initial du système.

La démarche précédente d'exploration preuve peut alors être appliquée simultanément ou consécutivement (ou simultanément à l'aide d'autant d'unité de calcul) à tous les itinéraires du poste. De cette manière, un certain nombre de combinaisons seront traitées plusieurs fois, assurant un recouvrement des preuves et par là, une couverture de l'ensemble des possibilités du poste.

Die allgemeine funktionelle Struktur der Signaltechnikprinzipien (allgemeine Graphen) ermöglicht es, die Abhängigkeiten zwischen den Fahrstraßen des Stellwerkes zu identifizieren. Die externen Eingänge, die keinen Graphen in Abhängigkeit mit denjenigen der Fahrstraße betreffen, werden im anfänglichen Systemszustand fixiert.

Die schon vorgestellte Methode des „Auswertungsbeweises" kann dann gleichzeitig oder aufeinanderfolgend (oder gleichzeitig mit mehreren Rechnereinheiten) auf alle Fahrstraßen des Stellwerks angewandt werden. Auf diese Weise werden bestimmte Kombinationen mehrmals behandelt was eine Überlappung der Beweise und so eine Abdeckung aller Möglichkeiten des Stellwerks garantiert.

## 7.4 Passage à niveau informatique

### 7.4.1 Programme du passage à niveau

La figure 7.32 décrit le fonctionnel d'un passage à niveau à signalisation automatique à deux demi barrières (SAL2). Celui-ci est installé sur ligne équipée en block automatique lumineux permissif (BAL) et en installation permanente de contre sens (IPCS).

Le déclenchement d'une annonce s'opère au franchissement du premier détecteur d'annonce de la voie circulée. Une fois le détecteur relâché, l'annonce est maintenue (continuité d'annonce) jusqu'à ce que la tête de la circulation atteigne le passage à niveau, puis que sa queue le libère. Les signaux routiers sont alors effacés si par ailleurs les autres annonces sont alors libres.

Afin de gérer les modes dégradés (libération tardive des détecteurs, libération tardive du circuit de voie assurant la continuité d'annonce), l'annonce doit être libérée automatiquement après un délai de 60s à compter du rétablissement de l'ensemble des conditions.

## 7.4 Elektronischer Bahnübergang

### 7.4.1 BÜ-Funktionen

Die Abbildung 7.32 beschreibt die Funktionsweise eines zweigleisigen Bahnübergangs mit automatischer Signalisierung und zwei Halbschranken (SAL2). Dieser befindet sich auf einer Strecke mit Selbstblock (BAL) und Lichtsignalen und einer ständigen Anlage zum Befahren des Gegengleises (IPCS).

Die automatische Zugvormeldung wird durch Überfahren des ersten Sensors der betroffenen Strecke aktiviert. Auch wenn der Zug den Sensor freigibt, wird die Vormeldung aufrechterhalten (Kontinuität der Vormeldung) bis die Zugspitze den Bahnübergang erreicht hat und das Zugende diesen freigibt. Die Signalgeber gehen aus, sobald auch die anderen Vormelder freigegeben sind.
Um auch die Rückfallebene zu berücksichtigen (verspätete Freigabe der Sensoren, verspätete Freigabe des Gleisstromkreises der die Vormeldung aufrechterhält), muss die Vormeldung automatisch 60 s nach der Wiederherstellung der Bedingungen freigegeben werden.

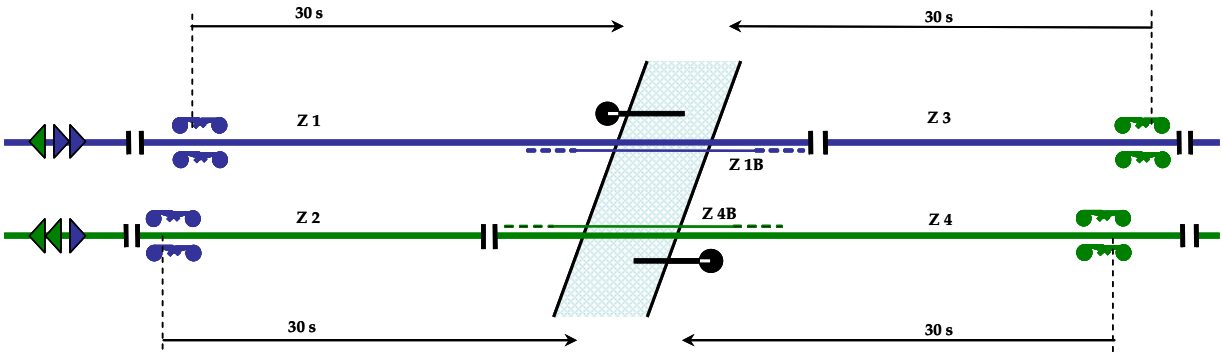



**Figure 7.32** : Description schématique du passage à niveau traité
**Abbildung 7.32**: Schematische Beschreibung des behandelten Bahnübergangs

Les fonctionnels des deux voies sont identiques et indépendants. Il en est de même des annonces d'une même voie.

Le programme schématique des annonces et le programme schématique de commande des signaux routiers sont décrits dans la figure 7.33.

Ces documents sont suffisants pour d'une part, décrire les graphes fonctionnels du passage à niveau et, d'autre part, pour définir les propriétés de sécurité que devra respecter le passage à niveau.

Die Funktionen der zwei Gleise sind identisch und unabhängig. Das Gleiche gilt für die Vormeldungen desselben Gleises.

Das Schema des Programms der Vormeldung und der Bedienung der Signalgeber ist in der Abbildung 7.33 beschrieben.

Diese Dokumente sind ausreichend um einerseits die funktionellen Graphen des Bahnübergangs zu erstellen und andererseits die Sicherheitseigenschaften zu definieren, die vom Bahnübergang eingehalten werden müssen.

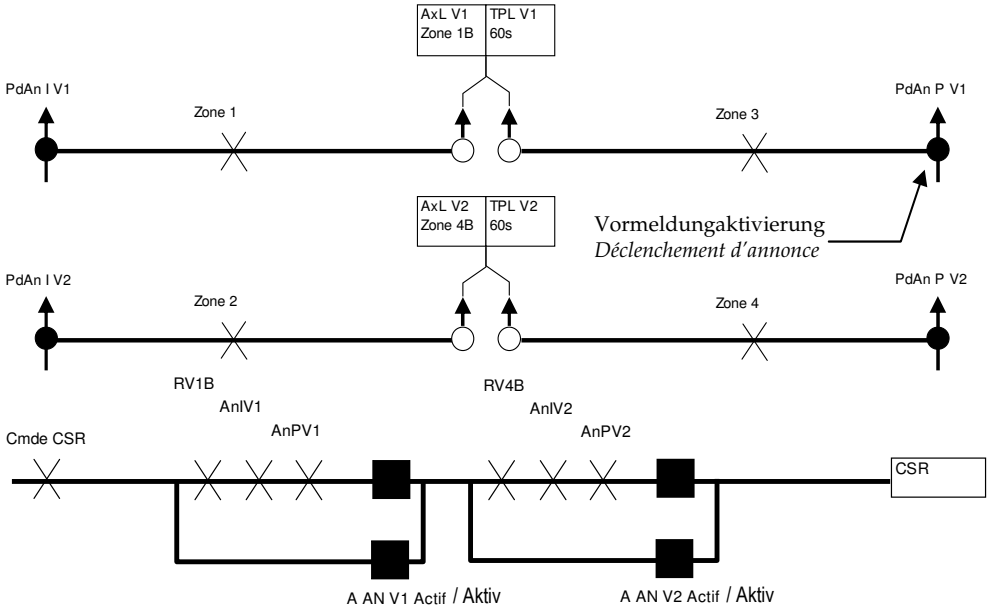



**Figure 7.33** : Programme schématique d'annonce et programme schématique de commande des signaux routiers du passage à niveau.
**Abbildung 7.33**: Schema des Programms der Vormeldung und der Steuerung der Signalgeber des Bahnübergangs.

## 7.4.2 Graphe fonctionnels en langage AEFD

La description des graphes fonctionnels a été réalisée dans l'optique d'une installation nouvelle.

### 7.4.2.1 Annonce au PN

Puisque les graphes dans les 4 sens sont similaires, seule l'annonce voie 1 sens pair sera détaillée.

Les graphes d'annonce sont au nombre de quatre (voie 1 sens pair, voie 1 sens impair, voie 2 sens pair, voie 2 sens impair). Une seule de ces 4 annonces occupée suffit à déclencher la commande des signaux routiers et l'abaissement des barrières (Figure 7.34).

## 7.4.2 Funktionellen Graphen in der AEFD-Sprache

Die Beschreibung der funktionellen Graphen ist im Hinblick auf eine neue Anlage durchgeführt worden.

### 7.4.2.1 Zugvormeldung am BÜ

Da die Graphen der vier Fahrtrichtungen identisch sind, wird nur die Vormeldung auf einem Gleis in einer Richtung detailliert dargestellt.

Es gibt vier Vormeldungsgraphen (Gleis 1 Fahrrichtung und Gegenrichtung und Gleis 2 Fahrrichtung und Gegenrichtung). Eine einzige Aktivierung genügt, um die Ampeln zu schalten und die Schranken zu schließen (Abb.7.34).

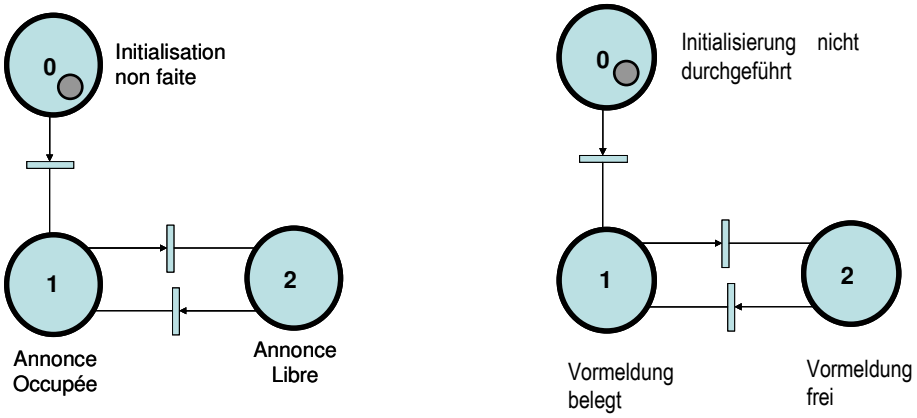



**Figure 7.34** : Graphe fonctionnel d'une annonce au PN
**Abbildung 7.34**: Funktioneller Graph für eine Zugvormeldung am BÜ

Ce qui peut s'écrire sous la forme du tableau suivante :

| *Automate* | **An** | | |
|---|---|---|---|
| *Place origine* | *0* | *Place destination* | *1* |
| *Événement* | **Initialisation** | | |
| *Condition* | **-** | | |
| *Action* | **Annonce occupée** | | |

| *Automate* | **An** | | |
|---|---|---|---|
| *Place origine* | *1* | *Place destination* | *2* |
| *Événement* | **Auxiliaire de libération actif OU condition de continuité d'annonce fausse OU condition de déclenchement d'annonce fausse** | | |
| *Condition* | **Auxiliaire de libération actif ET condition de continuité d'annonce fausse ET condition de déclenchement d'annonce fausse** | | |
| *Action* | **Annonce libre** | | |

| *Automate* | **An** | | |
|---|---|---|---|
| *Place origine* | *2* | *Place destination* | *1* |
| *Événement* | **Condition de déclenchement d'annonce vraie** | | |
| *Condition* | **-** | | |
| *Action* | **Annonce occupée** | | |

**Tableau 7.11** : Graphe fonctionnel d'une annonce au PN

Dies kann in der Form der Tabelle 7.11 geschrieben werden:

| *Automat* | **An** | | |
|---|---|---|---|
| *Ursprung* | *0* | *Ziel* | *1* |
| *Ereignis* | **Initialisierung** | | |
| *Bedingung* | **-** | | |
| *Handlung* | **Vormeldung belegt** | | |

| *Automat* | **An** | | |
|---|---|---|---|
| *Ursprung* | *1* | *Ziel* | *2* |
| *Ereignis* | **Hilfsmittel für die Auflösung aktiv ODER Bedingung für die Kontinuität der Vormeldung falsch ODER Bedingung zur Aktivierung der Vormeldung falsch** | | |
| *Bedingung* | **Hilfsmittel für die Auflösung aktiv UND Bedingung für die Kontinuität der Vormeldung falsch UND Bedingung zur Aktivierung der Vormeldung falsch** | | |
| *Handlung* | **Vormeldung frei** | | |

| *Automat* | **An** | | |
|---|---|---|---|
| *Ursprung* | *2* | *Ziel* | *1* |
| *Ereignis* | **Die Bedingung für die Aktivierung der Vormeldung ist wahr** | | |
| *Bedingung* | **-** | | |
| *Handlung* | **Vormeldung besetzt** | | |

**Tabelle 7.11**: Funktioneller Graph für eine Zugvormeldung am BÜ

### 7.4.2.2 Auxiliaire de libération

Le graphe de gestion de l'auxiliaire de libération (Figure 7.35) a pour rôle de permettre la libération de l'annonce une fois que la circulation a franchi le PN (rôle de la temporisation de libération). Il y a un automate AxL par voie.

### 7.4.2.2 Hilfsmittel für die Auflösung

Der Graph der Verwaltung der Hilfsmittel für die Auflösung (Abb. 7.35) hat die Rolle, die Vormeldung freizugeben sobald der Zug den Bahnübergang passiert hat (Rolle des Verzögerns der Auflösung). Es gibt ein Hilfsmittel für die Auflösung (AxL) pro Gleis.

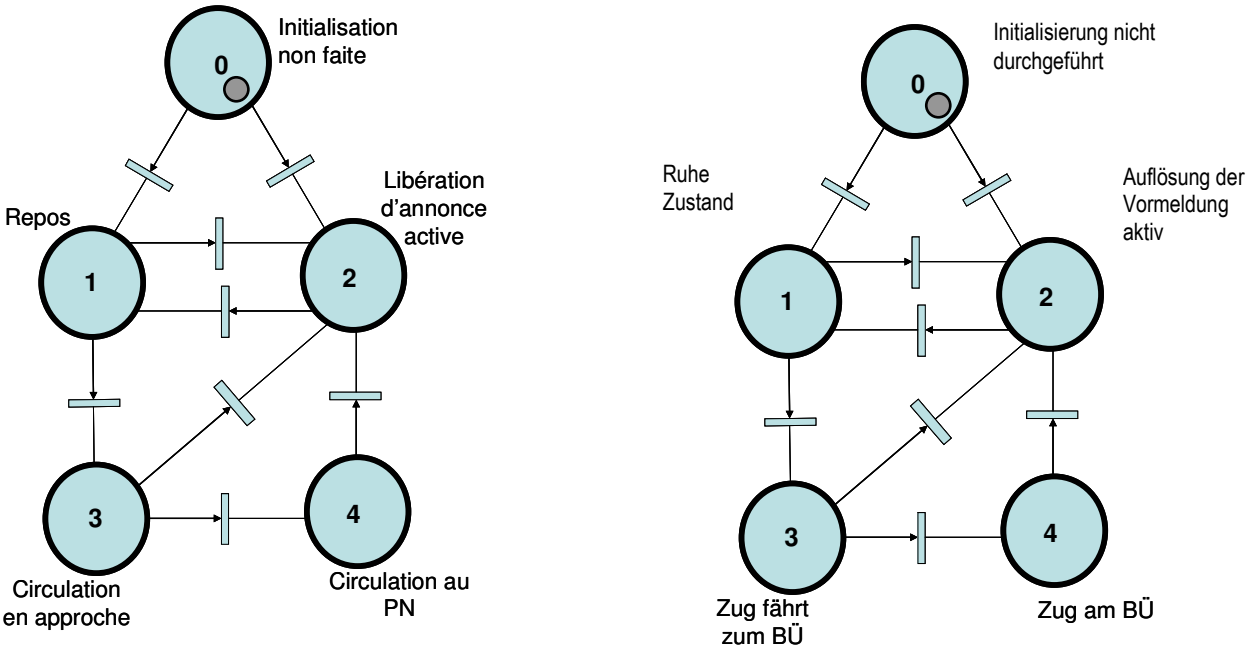



**Figure 7.35** : Graphe fonctionnel d'auxiliaire de libération
**Abbildung 7.35**: Funktioneller Graph der Hilfsmittel für die Auflösung

Ce qui peut s'écrire sous la forme du tableau 7.12.

| ***Automate*** | AxL | | |
|---|---|---|---|
| ***Place origine*** | 0 | ***Place destination*** | 1 |
| ***Événement*** | Initialisation | | |
| ***Condition*** | Zone courte libre | | |
| ***Action*** | Auxiliaire de libération inactif | | |

| ***Automate*** | AxL | | |
|---|---|---|---|
| ***Place origine*** | 0 | ***Place destination*** | 2 |
| ***Événement*** | Initialisation | | |
| ***Condition*** | Zone courte occupée | | |
| ***Action*** | Auxiliaire de libération actif | | |

| ***Automate*** | AxL | | |
|---|---|---|---|
| ***Place origine*** | 1 | ***Place destination*** | 2 |
| ***Événement*** | Fin de temporisation de libération | | |
| ***Condition*** | - | | |
| ***Action*** | Auxiliaire de libération actif | | |

| ***Automate*** | AxL | | |
|---|---|---|---|
| ***Place origine*** | 1 | ***Place destination*** | 3 |
| ***Événement*** | Une des annonces de la voie occupée OU une des zones de la voie occupée | | |
| ***Condition*** | Une annonce de la voie occupée ET sa zone de continuité d'annonce occupée (continuité d'annonce) | | |
| ***Action*** | - | | |

| ***Automate*** | AxL | | |
|---|---|---|---|
| ***Place origine*** | 2 | ***Place destination*** | 1 |
| ***Événement*** | Annonce de sens pair libérée OU annonce de sens impair libérée | | |
| ***Condition*** | Annonce de sens pair et impair libérées | | |
| ***Action*** | - | | |

| ***Automate*** | AxL | | |
|---|---|---|---|
| ***Place origine*** | 3 | ***Place destination*** | 2 |
| ***Événement*** | Fin de temporisation de libération | | |
| ***Condition*** | - | | |
| ***Action*** | Auxiliaire de libération actif | | |

| ***Automate*** | AxL | | |
|---|---|---|---|
| ***Place origine*** | 3 | ***Place destination*** | 4 |
| ***Événement*** | Occupation de la zone courte | | |
| ***Condition*** | - | | |
| ***Action*** | - | | |

| ***Automate*** | AxL | | |
|---|---|---|---|
| ***Place origine*** | 4 | ***Place destination*** | 2 |
| ***Événement*** | Libération de la continuité d'annonce | | |
| ***Condition*** | - | | |
| ***Action*** | Auxiliaire de libération actif | | |

**Tableau 7.12** : Description du graphe fonctionnel de la fonction AxL

Dies kann in Form der Tabelle 7.12 geschrieben werden.

| ***Automat*** | AxL | | |
|---|---|---|---|
| ***Ursprung*** | 0 | ***Ziel*** | 1 |
| ***Ereignis*** | Initialisierung | | |
| ***Bedingung*** | Kurze Zone frei | | |
| ***Handlung*** | Hilfsmittel für die Auflösung inaktiv | | |

| ***Automat*** | AxL | | |
|---|---|---|---|
| ***Ursprung*** | 0 | ***Ziel*** | 2 |
| ***Ereignis*** | Initialisierung | | |
| ***Bedingung*** | Kurze Zone besetzt | | |
| ***Handlung*** | Hilfsmittel für die Auflösung aktiv | | |

| ***Automat*** | AxL | | |
|---|---|---|---|
| ***Ursprung*** | 1 | ***Ziel*** | 2 |
| ***Ereignis*** | Ende der Verzögerung der Auflösung | | |
| ***Bedingung*** | - | | |
| ***Handlung*** | Hilfsmittel für die Auflösung aktiv | | |

| ***Automat*** | AxL | | |
|---|---|---|---|
| ***Ursprung*** | 1 | ***Ziel*** | 3 |
| ***Ereignis*** | Eine Vormeldung belegt ODER ein Gleisstromkreis belegt | | |
| ***Bedingung*** | Eine Vormeldung belegt UND die Zone der Kontinuität der Vormeldung belegt (Kontinuität der Vormeldung) | | |
| ***Handlung*** | - | | |

| ***Automat*** | AxL | | |
|---|---|---|---|
| ***Ursprung*** | 2 | ***Ziel*** | 1 |
| ***Ereignis*** | Vormeldung der Fahrrichtung ODER Vormeldung der Gegenrichtung frei | | |
| ***Bedingung*** | Vormeldung der Fahrrichtung UND Vormeldung der Gegenrichtung frei | | |
| ***Handlung*** | - | | |

| ***Automat*** | AxL | | |
|---|---|---|---|
| ***Ursprung*** | 3 | ***Ziel*** | 2 |
| ***Ereignis*** | Ende der Verzögerung der Auflösung | | |
| ***Bedingung*** | - | | |
| ***Handlung*** | Hilfsmittel für die Auflösung aktiv | | |

| ***Automat*** | AxL | | |
|---|---|---|---|
| ***Ursprung*** | 3 | ***Ziel*** | 4 |
| ***Ereignis*** | Besetzung des kurzen Gleisstromkreises | | |
| ***Bedingung*** | - | | |
| ***Handlung*** | - | | |

| ***Automat*** | AxL | | |
|---|---|---|---|
| ***Ursprung*** | 4 | ***Ziel*** | 2 |
| ***Ereignis*** | Freigeben der Kontinuität der Vormeldung | | |
| ***Bedingung*** | - | | |
| ***Handlung*** | Hilfsmittel für die Auflösung aktiv | | |

**Tabelle 7.12**:Beschreibung des funktionellen Graphen der AxL-Funktion

### 7.4.2.3 Temporisation de libération

La temporisation de libération permet de libérer une annonce lorsqu'il s'est écoulé un temps entre le déclenchement de l'annonce et la disparition des conditions de déclenchement et de continuité de l'annonce. Elle intervient donc lorsqu'une circulation déclenche une annonce et rebrousse chemin avant le PN. Il y a un graphe par voie.

### 7.4.2.3 Verzögerung der Auflösung

Der Graph für die Verzögerung der Auflösung ermöglicht es, eine Vormeldung aufzulösen, wenn eine bestimmte Zeit zwischen der Aktivierung der Vormeldung und dem Wegfall der Bedingungen für die Kontinuität der Vormeldung verstrichen ist. Die Verzögerung tritt also dann in Aktion, wenn ein Zug die Vormeldung aktiviert und vor dem Bahnübergang umkehrt. Es existiert ein Graph pro Gleis (Abb. 7.36).

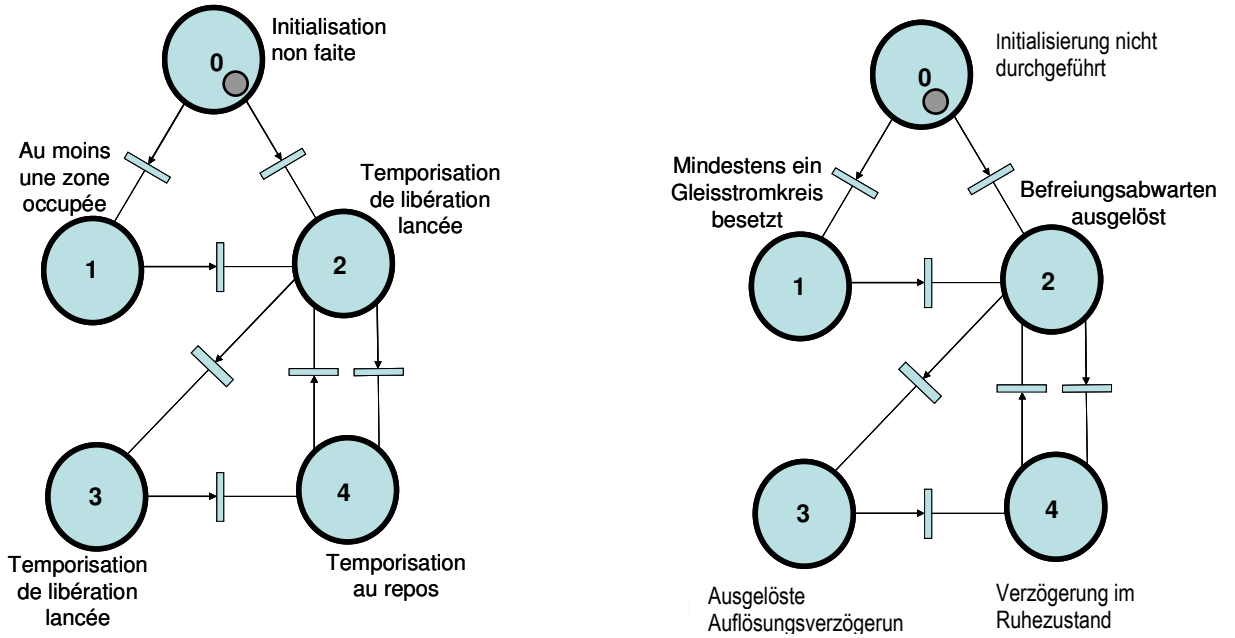



**Figure 7.36** : Graphe fonctionnel du graphe de temporisation de libération (TPL)
**Abbildung 7.36**: Funktionelle Graph der Verzögerungsauflösung (TPL)

Ce qui peut s'écrire sous la forme suivante :

| *Automate* | TPL | | |
|---|---|---|---|
| *Place origine* | 0 | *Place dest.* | 1 |
| *Événement* | Initialisation | | |
| *Condition* | Une zone occupée | | |
| *Action* | - | | |

| *Automate* | TPL | | |
|---|---|---|---|
| *Place origine* | 0 | *Place dest.* | 2 |
| *Événement* | Initialisation | | |
| *Condition* | Les deux zones de libre | | |
| *Action* | Début de temporisation de libération | | |

| *Automate* | TPL | | |
|---|---|---|---|
| *Place origine* | 2 | *Place dest.* | 3 |
| *Événement* | Fin de temporisation de libération | | |
| *Condition* | - | | |
| *Action* | Mémorisation de fin de temporisation | | |

| *Automate* | TPL | | |
|---|---|---|---|
| *Place origine* | 2 | *Place dest.* | 4 |
| *Événement* | Libération d'une annonce OU occupation d'une zone | | |
| *Condition* | Annonce libre pour les deux sens OU occupation d'une zone | | |
| *Action* | Arrêt de la temporisation | | |

Dies kann in Form der Tabelle 7.13 geschrieben werden:

| *Automat* | TPL | | |
|---|---|---|---|
| *Ursprung* | 0 | *Ziel* | 1 |
| *Ereignis* | Initialisierung | | |
| *Bedingung* | Ein Gleisstromkreis besetzt | | |
| *Handlung* | - | | |

| *Automat* | TPL | | |
|---|---|---|---|
| *Ursprung* | 0 | *Ziel* | 2 |
| *Ereignis* | Initialisierung | | |
| *Bedingung* | Beide Gleisstromkreise sind frei | | |
| *Handlung* | Anfang der Auflösungsverzögerung | | |

| *Automat* | TPL | | |
|---|---|---|---|
| *Ursprung* | 2 | *Ziel* | 3 |
| *Ereignis* | Ende des Auflösungsverzögerung | | |
| *Bedingung* | - | | |
| *Handlung* | Speicherung des Verzögerungsendes | | |

| *Automate* | TPL | | |
|---|---|---|---|
| *Ursprung* | 2 | *Ziel* | 4 |
| *Ereignis* | Auflösung einer Vormeldung ODER belegter Gleisstromkreis | | |
| *Bedingung* | Vormeldung frei in beiden Richtungen ODER belegter Gleisstromkreis | | |
| *Handlung* | Ende der Verzögerung | | |

| *Automate* | TPL | | |
|---|---|---|---|
| *Place origine* | 3 | *Place dest.* | 4 |
| *Événement* | Libération d'une annonce OU occupation d'une zone | | |
| *Condition* | Annonce libre pour les deux sens OU occupation d'une zone | | |
| *Action* | - | | |

| *Automate* | TPL | | |
|---|---|---|---|
| *Place origine* | 4 | *Place dest.* | 2 |
| *Événement* | Une annonce occupée OU une zone libérée | | |
| *Condition* | Une annonce occupée ET les deux zones libres | | |
| *Action* | Début de temporisation | | |

**Tableau 7.13** : Graphe fonctionnel du graphe de temporisation de libération (TPL)

| *Automat* | TPL | | |
|---|---|---|---|
| *Ursprung* | 3 | *Ziel* | 4 |
| *Ereignis* | Auflösung einer Vormeldung ODER Belegung eines Gleisstromkreises | | |
| *Bedingung* | Vormeldung frei in beiden Richtungen ODER Belegung eines Gleisstromkreises | | |
| *Handlung* | - | | |

| *Automat* | TPL | | |
|---|---|---|---|
| *Ursprung* | 4 | *Ziel* | 2 |
| *Ereignis* | Eine Vormeldung belegt ODER ein Gleisstromkreis frei | | |
| *Bedingung* | Eine Vormeldung belegt UND beide Gleisstromkreis frei | | |
| *Handlung* | Anfang der Verzögerung | | |

**Tabelle 7.13**: Funktioneller Graph der Verzögerungs-auflösung (TPL)

### 7.4.2.4 Commande des signaux routiers

L'automate de commande des signaux routiers (Fig. 7.37) est en charge de la « centralisation » de l'état des quatre annonces et du déclenchement, si il y a lieu, de l'allumage des signaux via la commande d'un relais (sortie terrain).

### 7.4.2.4 Steuerung der Signalgeber

Der Automat der Steuerung der Ampeln (Abb. 7.37) hat die Aufgabe, den Zustand der vier Vormeldungen zu zentralisieren und, falls eine davon belegt ist, die Ampeln über eine Relaissteuerung einzuschalten (vor Ort Ausgang).

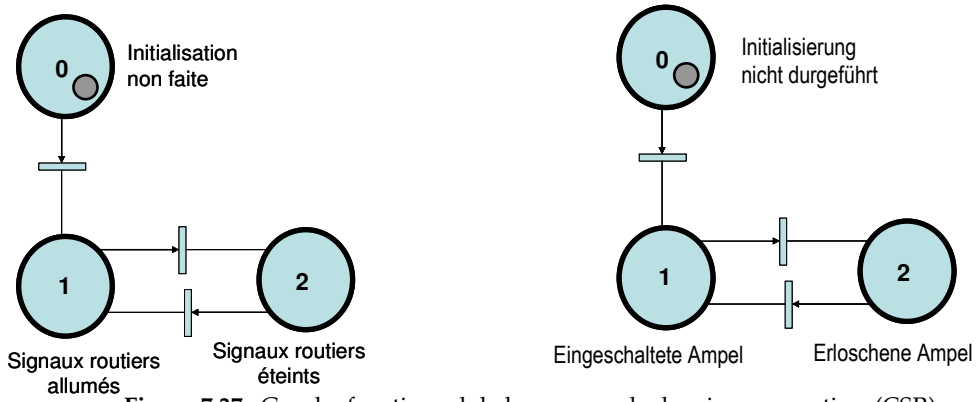



**Figure 7.37** : Graphe fonctionnel de la commande des signaux routiers (CSR)
**Abbildung 7.3** : Funktioneller Graph für die Ampelsteuerung (CSR)

| *Automate* | CSR | | |
|---|---|---|---|
| *Place origine* | 0 | *Place dest.* | 1 |
| *Événement* | Initialisation | | |
| *Condition* | | | |
| *Action* | Commande du relais de sortie à la chute | | |

| *Automate* | CSR | | |
|---|---|---|---|
| *Place origine* | 1 | *Place dest.* | 2 |
| *Événement* | Une annonce de libre OU une zone courte de libre | | |
| *Condition* | Toutes les annonces libres ET toutes les zones courtes libres | | |
| *Action* | Commande du relais de sortie à l'excitation | | |

| *Automate* | CSR | | |
|---|---|---|---|
| *Place origine* | 2 | *Place dest.* | 1 |
| *Événement* | Une annonce de occupée OU une zone courte occupée | | |
| *Condition* | - | | |
| *Action* | Commande du relais de sortie à la chute | | |

**Tableau 7.14** : Graphe fonctionnel de commande des signaux routiers (CSR)

| *Automat* | CSR | | |
|---|---|---|---|
| *Ursprung* | 0 | *Ziel* | 1 |
| *Ereignis* | Initialisierung | | |
| *Bedingung* | | | |
| *Handlung* | Steuerung des Falls des Ausgangsrelais | | |

| *Automat* | CSR | | |
|---|---|---|---|
| *Ursprung* | 1 | *Ziel* | 2 |
| *Ereignis* | Eine Vormeldung frei ODER ein kurzer Gleisstromkreis frei | | |
| *Bedingung* | Alle Vormeldungen frei UND alle kurzen Gleisstromkreise frei | | |
| *Handlung* | Befehl des Hochstellens des Ausgangsrelais | | |

| *Automat* | CSR | | |
|---|---|---|---|
| *Ursprung* | 2 | *Ziel* | 1 |
| *Ereignis* | Eine belegte Vormeldung ODER eine kurzer Gleisstromkreis belegt | | |
| *Bedingung* | - | | |
| *Handlung* | Befehl des Falls des Ausgangsrelais | | |

**Tabelle 7.14**: Funktioneller Graph für die Ampelsteuerung (CSR)

## 7.4.3 Propriétés de sécurité et postulats

### 7.4.3.1 Initialisation

L'automate de preuve AP_INIT interdit la modification de toute entrée terrain tant que la phase d'initialisation du PN n'est pas achevée. Il empêche également l'exploration des cas où une annonce est déclenchée alors que l'annonce de sens contraire est déjà occupée (correspondant au cas où deux trains rouleraient l'un vers l'autre). Enfin il neutralise les entrées correspondant au commutateur de mise en marche une fois le système mis en fonctionnement (réglementation). Ces précautions réduisent significativement le temps de calcul.

## 7.4.3 Sicherheitseigenschaften und Anforderungen

### 7.4.3.1 Initialisierung

Der Beweisautomat AP_INIT verbietet die Änderung eines vor Ort Eingangs solange die Initialisierungsphase nicht abgeschlossen ist. Er verhindert auch die Auswertung der Fälle, in denen eine Vormeldung ausgelöst wird obwohl die Vormeldung in der anderen Richtung schon belegt ist (das entspricht dem Fall, in dem zwei Zügen aufeinander zu fahren). Letztendlich neutralisiert er, sobald das System hochgefahren ist, die Eingänge die vom Anschalten herrühren (Vorschriften). Diese Vorsichtsmaßnahmen reduzieren die Rechenzeit maßgeblich.

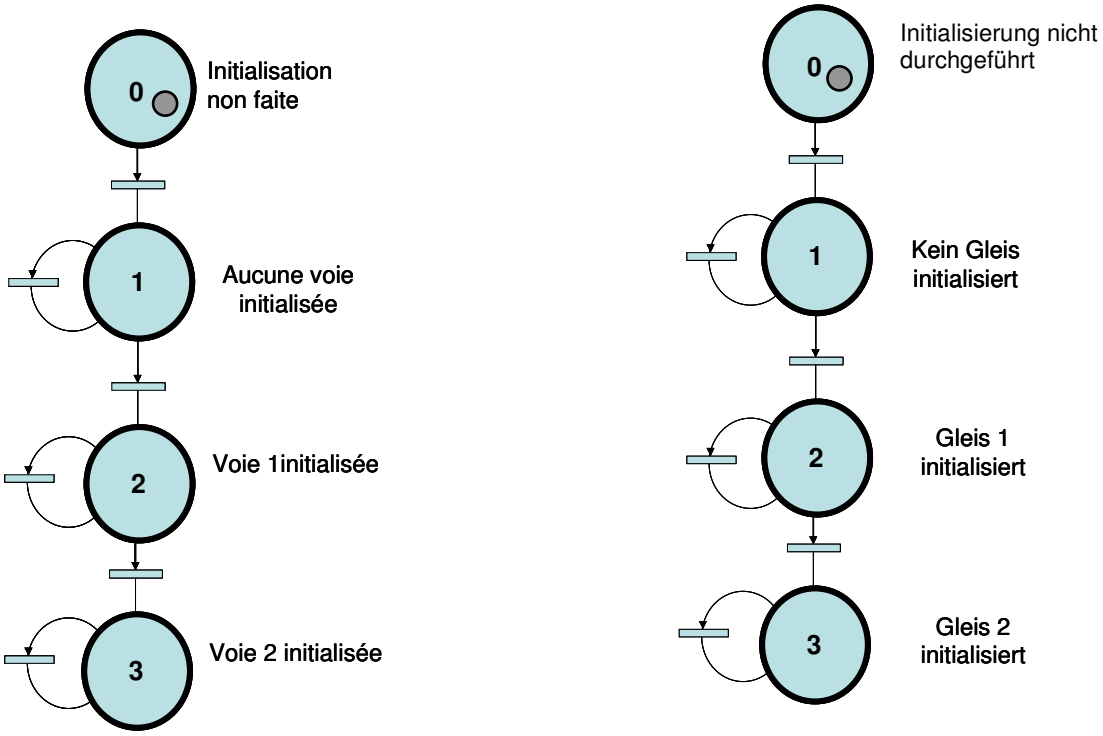



**Figure 7.38** : Graphe fonctionnel de description des postulats de fonctionnement (AP_INIT)
**Abbildung 7.38**: Funktioneller Graph für die Beschreibung der Funktionsanforderungen (AP_INIT)

| *Automate* | AP_INIT | | |
|---|---|---|---|
| *Place origine* | 0 | *Place dest.* | 1 |
| *Événement* | Initialisation | | |
| *Condition* | - | | |
| *Action* | - | | |

| *Automat* | AP_INIT | | |
|---|---|---|---|
| *Ursprung* | 0 | *Ziel* | 1 |
| *Ereignis* | Initialisierung | | |
| *Bedingung* | - | | |
| *Handlung* | - | | |

| *Automate* | AP_INIT | | |
|---|---|---|---|
| *Place origine* | 1 | *Place dest.* | 1 |
| *Événement* | Tous les événements externes sauf fin de temporisation de libération de la voie 1 | | |
| *Condition* | - | | |
| *Action* | P_6 | | |

| *Automat* | AP_INIT | | |
|---|---|---|---|
| *Ursprung* | 1 | *Ziel* | 1 |
| *Ereignis* | Alle externen Ereignisse außer dem Ende der Auflösungsverzögerung auf Gleis 1 | | |
| *Bedingung* | - | | |
| *Handlung* | P_6 | | |

| *Automate* | AP_INIT | | |
|---|---|---|---|
| *Place origine* | 1 | *Place dest.* | 2 |
| *Événement* | Fin de temporisation de libération de la voie 1 | | |
| *Condition* | - | | |
| *Action* | - | | |

| *Automate* | AP_INIT | | |
|---|---|---|---|
| *Place origine* | 2 | *Place dest.* | 2 |
| *Événement* | Tous les événements externes sauf la fin de temporisation de libération de la voie 2 | | |
| *Condition* | - | | |
| *Action* | P_6 | | |

| *Automate* | AP_INIT | | |
|---|---|---|---|
| *Place origine* | 2 | *Place dest.* | 3 |
| *Événement* | Fin de temporisation de libération de la voie 2 | | |
| *Condition* | - | | |
| *Action* | - | | |

| *Automate* | AP_INIT | | |
|---|---|---|---|
| *Place origine* | 3 | *Place dest.* | 3 |
| *Événement* | Occupation d'une annonce ou attaque d'une pédale | | |
| *Condition* | Attaque d'une pédale qui déclencherait une annonce de sens contraire à une annonce déjà occupée | | |
| *Action* | P_6 | | |

| *Automate* | AP_INIT | | |
|---|---|---|---|
| *Place origine* | 3 | *Place dest.* | 3 |
| *Événement* | Modification de l'état du commutateur de mise en service | | |
| *Condition* | - | | |
| *Action* | P_6 | | |

| *Automate* | AP_P6_CDV | | |
|---|---|---|---|
| *Place origine* | 0 | *Place dest.* | 1 |
| *Événement* | Initialisation | | |
| *Condition* | - | | |
| *Action* | - | | |

| *Automate* | AP_P6_CDV | | |
|---|---|---|---|
| *Place origine* | 1 | *Place dest.* | 1 |
| *Événement* | Zone courte occupée | | |
| *Condition* | Zone dans laquelle la zone courte est incluse libre | | |
| *Action* | P_6 | | |

**Tableau 7.15** : Graphe fonctionnel de description des postulats de fonctionnement (AP_INIT)

## 7.4.3.2 Vérification du comportement d'une annonce

L'automate AP_An vérifie qu'une situation contraire à la sécurité ne peut se produire :

- jamais une annonce ne reste libre si la condition de déclenchement d'annonce est vraie ;
- jamais une annonce n'est libérée alors que les conditions de libération ne sont pas remplies.

| *Automat* | AP_INIT | | |
|---|---|---|---|
| *Ursprung* | 1 | *Ziel* | 2 |
| *Ereignis* | Ende der Auflösungsverzögerung auf Gleis 1 | | |
| *Bedingung* | - | | |
| *Handlung* | - | | |

| *Automat* | AP_INIT | | |
|---|---|---|---|
| *Ursprung* | 2 | *Ziel* | 2 |
| *Ereignis* | Alle externen Ereignisse außer dem Ende der Auflösungsverzögerung auf Gleis 2 | | |
| *Bedingung* | - | | |
| *Handlung* | P_6 | | |

| *Automat* | AP_INIT | | |
|---|---|---|---|
| *Ursprung* | 2 | *Ziel* | 3 |
| *Ereignis* | Ende der Auflösungsverzögerung auf Gleis 2 | | |
| *Bedingung* | - | | |
| *Handlung* | - | | |

| *Automat* | AP_INIT | | |
|---|---|---|---|
| *Ursprung* | 3 | *Ziel* | 3 |
| *Ereignis* | Belegung einer Vormeldung ODER Betätigen eines mechanischen Gleisschaltmittels | | |
| *Bedingung* | Betätigen eines mechanischen Gleisschaltmittels das eine Vormeldung in der Gegenrichtung zu einer bestehenden Vormeldung auslösen würde | | |
| *Handlung* | P_6 | | |

| *Automat* | AP_INIT | | |
|---|---|---|---|
| *Ursprung* | 3 | *Ziel* | 3 |
| *Ereignis* | Veränderung des Inbetriebnahmeschalters | | |
| *Bedingung* | - | | |
| *Handlung* | P_6 | | |

| *Automat* | AP_P6_CDV | | |
|---|---|---|---|
| *Ursprung* | 0 | *Ziel* | 1 |
| *Ereignis* | Initialisierung | | |
| *Bedingung* | - | | |
| *Handlung* | - | | |

| *Automat* | AP_P6_CDV | | |
|---|---|---|---|
| *Ursprung* | 1 | *Ziel* | 1 |
| *Ereignis* | Kurzer Gleisstromkreis belegt | | |
| *Bedingung* | Gleisstromkreis, der den kurzen Gleisstromkreis umfasst, frei | | |
| *Handlung* | P_6 | | |

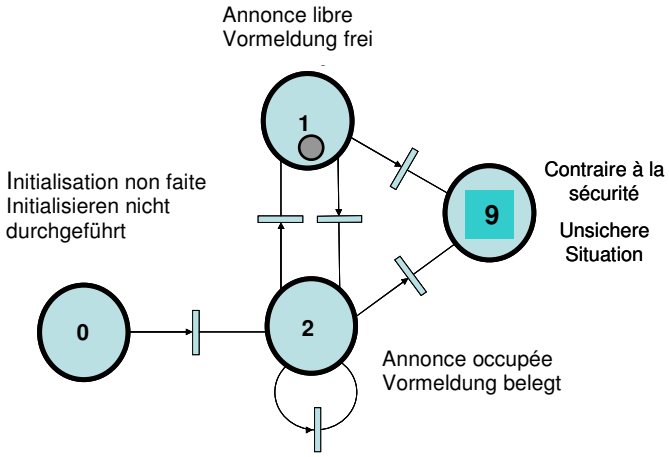
**Tabelle 7.15**: Funktioneller Graph für die Beschreibung der Funktionsanforderungen (AP_INIT)

## 7.4.3.2 Überprüfung des Verhaltens einer Vormeldung

Der Beweisautomat AP_An überprüft, dass keine unsichere Situation auftritt:

- Eine Vormeldung bleibt niemals frei, wenn die Bedingung für das Auslösen der Vormeldung wahr ist.
- Eine Vormeldung wird niemals aufgelöst wenn die Auflösungsbedingungen nicht erfüllt sind.

Il s'assure d'autre part que jamais l'annonce reste occupée alors qu'elle devrait être libérée. Il y a un automate de preuve (AP) par annonce.




Auf der anderen Seite vergewissert er sich, dass die Vormeldung niemals belegt bleibt, wenn sie aufgelöst werden müsste. Es existiert ein Automat pro Vormeldung.

**Figure 7.39** : Graphe de preuve de vérification d'une annonce pour une voie et un sens donné (AP_An)
**Abbildung 7.39**: Beweisgraph für die Überprüfung einer Vormeldung für ein Gleis und eine Richtung (AP_An)

| ***Automate*** | AP_An | | |
|---|---|---|---|
| ***Place origine*** | 0 | ***Place dest.*** | 2 |
| ***Événement*** | Initialisation | | |
| ***Condition*** | - | | |
| ***Action*** | - | | |

| ***Automate*** | AP_An | | |
|---|---|---|---|
| ***Place origine*** | 1 | ***Place dest.*** | 2 |
| ***Événement*** | Condition de déclenchement d'annonce OU annonce occupée | | |
| ***Condition*** | Condition de déclenchement d'annonce ET annonce occupée | | |
| ***Action*** | - | | |

| ***Automate*** | AP_An | | |
|---|---|---|---|
| ***Place origine*** | 1 | ***Place dest.*** | 9 |
| ***Événement*** | Condition de déclenchement d'annonce OU annonce libre | | |
| ***Condition*** | Condition de déclenchement d'annonce ET annonce libre | | |
| ***Action*** | P_5 | | |

| ***Automate*** | AP_An | | |
|---|---|---|---|
| ***Place origine*** | 2 | ***Place dest.*** | 1 |
| ***Événement*** | Condition de libération d'annonce OU annonce libre | | |
| ***Condition*** | Condition de libération d'annonce ET annonce libre | | |
| ***Action*** | - | | |

| ***Automate*** | AP_An | | |
|---|---|---|---|
| ***Place origine*** | 2 | ***Place dest.*** | 2 |
| ***Événement*** | Condition de libération d'annonce OU annonce occupée | | |
| ***Condition*** | Condition de libération d'annonce ET annonce occupée | | |
| ***Action*** | P_7 | | |

| ***Automate*** | AP_An | | |
|---|---|---|---|
| ***Place origine*** | 2 | ***Place dest.*** | 9 |
| ***Événement*** | NON (Condition de libération d'annonce) OU annonce libre | | |
| ***Condition*** | NON (Condition de libération d'annonce) ET annonce libre | | |
| ***Action*** | P_5 | | |

**Tableau 7.16** : Graphe de preuve de vérification d'une annonce pour une voie et un sens donné (AP_An)

| ***Automat*** | AP_An | | |
|---|---|---|---|
| ***Ursprung*** | 0 | ***Ziel*** | 2 |
| ***Ereignis*** | Initialisierung | | |
| ***Bedingung*** | - | | |
| ***Handlung*** | - | | |

| ***Automat*** | AP_An | | |
|---|---|---|---|
| ***Ursprung*** | 1 | ***Ziel*** | 2 |
| ***Ereignis*** | Bedingungen für das Auslösen einer Vormeldung ODER belegte Vormeldung | | |
| ***Bedingung*** | Bedingung für das Auslösen einer Vormeldung UND belegte Vormeldung | | |
| ***Handlung*** | - | | |

| ***Automat*** | AP_An | | |
|---|---|---|---|
| ***Ursprung*** | 1 | ***Ziel*** | 9 |
| ***Ereignis*** | Bedingungen für das Auslösen einer Vormeldung ODER freie Vormeldung | | |
| ***Bedingung*** | Bedingung für das Auslösen einer Vormeldung UND freie Vormeldung | | |
| ***Handlung*** | P_5 | | |

| ***Automat*** | AP_An | | |
|---|---|---|---|
| ***Ursprung*** | 2 | ***Ziel*** | 1 |
| ***Ereignis*** | Bedingung für die Auflösung der Vormeldung ODER freie Vormeldung | | |
| ***Bedingung*** | Bedingung für die Auflösung der Vormeldung UND freie Vormeldung | | |
| ***Handlung*** | - | | |

| ***Automat*** | AP_An | | |
|---|---|---|---|
| ***Ursprung*** | 2 | ***Ziel*** | 2 |
| ***Ereignis*** | Bedingung für dir Auflösung der Vormeldung ODER belegte Vormeldung | | |
| ***Bedingung*** | Bedingung für dir Auflösung der Vormeldung UND belegte Vormeldung | | |
| ***Handlung*** | P_7 | | |

| ***Automat*** | AP_An | | |
|---|---|---|---|
| ***Ursprung*** | 2 | ***Ziel*** | 9 |
| ***Ereignis*** | NICHT (Bedingung für die Auflösung der Vormeldung) ODER freie Vormeldung | | |
| ***Bedingung*** | NICHT (Bedingung für die Auflösung der Vormeldung) UND freie Vormeldung | | |
| ***Handlung*** | P_5 | | |

**Tabelle 7.16**: Beweisgraph für die Überprüfung einer Vormeldung für ein Gleis und eine Richtung (AP_An)

### 7.4.3.3 Contrôle de séquence de réarmement

L'automate AP_Rea (Fig. 7.40) mémorise qu'une séquence de réarmement a eu lieu (occupation de la zone courte suivie de libération de la zone de continuité d'annonce). Cet indicateur est utilisé par AP_An.

### 7.4.3.3 Die Kontrolle der Reaktivierung

Der Beweisautomat AP_Rea (Abb. 7.40) speichert, dass eine Reaktivierungssequenz stattgefunden hat (Belegung Gleisstromkreises gefolgt von der Auflösung des Gleisstromkreises der Vormeldekontinuität). Dieser Indikator wird von AP_An benützt.

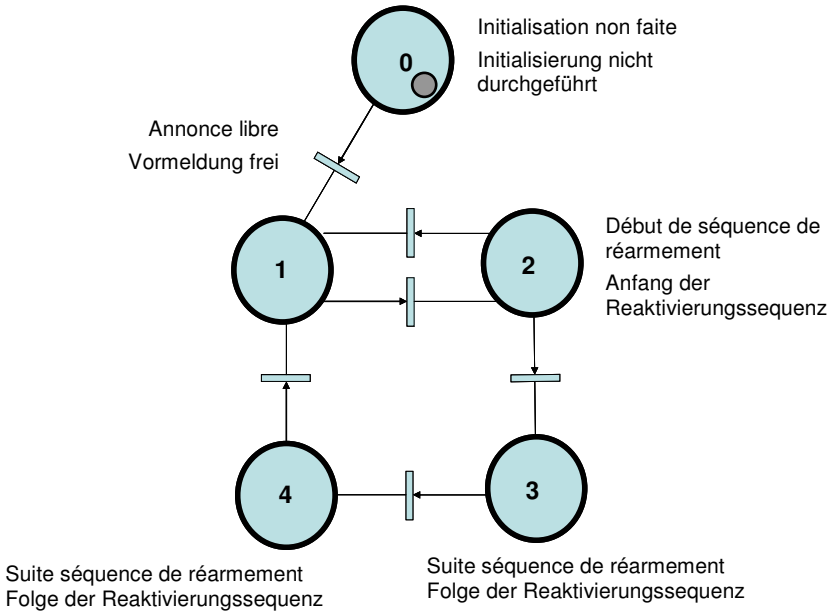



**Figure 7.40** : Graphe de preuve de vérification de la séquence de réarmement pour une voie et un sens donné (AP_Rea)
**Abbildung 7.40**: Beweisgraph für die Überprüfung einer Reaktivierungssequenz für ein Gleis und eine Richtung (AP_Rea)

| *Automate* | AP_Rea | | |
|---|---|---|---|
| *Place origine* | 0 | *Place destination* | 1 |
| *Événement* | Initialisation | | |
| *Condition* | - | | |
| *Action* | - | | |

| *Automate* | AP_Rea | | |
|---|---|---|---|
| *Place origine* | 1 | *Place destination* | 2 |
| *Événement* | Occupation d'une annonce OU occupation d'une zone | | |
| *Condition* | Occupation d'une annonce ET de la zone correspondant à la condition de continuité | | |
| *Action* | - | | |

| *Automate* | AP_Rea | | |
|---|---|---|---|
| *Place origine* | 2 | *Place destination* | 1 |
| *Événement* | Libération d'une annonce | | |
| *Condition* | - | | |
| *Action* | - | | |

| *Automate* | AP_Rea | | |
|---|---|---|---|
| *Place origine* | 2 | *Place destination* | 3 |
| *Événement* | Occupation de la zone courte | | |
| *Condition* | - | | |
| *Action* | - | | |

| *Automate* | AP_Rea | | |
|---|---|---|---|
| *Place origine* | 3 | *Place destination* | 4 |
| *Événement* | Libération d'une zone | | |
| *Condition* | Une annonce occupée ET libération de la zone correspondant à la continuité d'annonce | | |
| *Action* | Mémorisation du fait que la séquence de réarmement d'annonce a été effectuée | | |

| *Automat* | AP_Rea | | |
|---|---|---|---|
| *Ursprung* | 0 | *Ziel* | 1 |
| *Ereignis* | Initialisierung | | |
| *Bedingung* | - | | |
| *Handlung* | - | | |

| *Automat* | AP_Rea | | |
|---|---|---|---|
| *Ursprung* | 1 | *Ziel* | 2 |
| *Ereignis* | Vormeldung belegt ODER Gleisstromkreis belegt | | |
| *Bedingung* | Belegung der Vormeldung UND des Gleisstromkreises der zur Vormeldekontinuität gehört | | |
| *Handlung* | - | | |

| *Automat* | AP_Rea | | |
|---|---|---|---|
| *Ursprung* | 2 | *Ziel* | 1 |
| *Ereignis* | Auflösung einer Vormeldung | | |
| *Bedingung* | - | | |
| *Handlung* | - | | |

| *Automat* | AP_Rea | | |
|---|---|---|---|
| *Ursprung Platz* | 2 | *Ziel* | 3 |
| *Ereignis* | Belegung eines kurzen Gleisstromkreis | | |
| *Bedingung* | - | | |
| *Handlung* | - | | |

| *Automat* | AP_Rea | | |
|---|---|---|---|
| *Ursprung* | 3 | *Ziel* | 4 |
| *Ereignis* | Freimachen eines Gleisstromkreis | | |
| *Bedingung* | Eine belegte Vormeldung UND Freimachen des Gleisstromkreises, der zur Vormeldekontinuität gehört | | |
| *Handlung* | Speicherung der Tatsache, dass die Reaktivierungssequenz ausgeführt wurde | | |

| *Automate* | AP_Rea | | |
|---|---|---|---|
| *Place origine* | 4 | *Place destination* | 1 |
| *Événement* | Libération d'une annonce | | |
| *Condition* | - | | |
| *Action* | - | | |

**Tableau 7.17 :** Graphe de preuve de vérification de la séquence de réarmement pour une voie et un sens donné (AP_Rea)

| *Automat* | AP_Rea | | |
|---|---|---|---|
| *Ursprung* | 4 | *Ziel* | 1 |
| *Ereignis* | Auflösung einer Vormeldung | | |
| *Bedingung* | - | | |
| *Handlung* | - | | |

**Tabelle 7.17**: Beweisgraph für die Überprüfung einer Reaktivierungssequenz für ein Gleis und eine Richtung (AP_Rea)

### 7.4.3.4 Contrôle de séquence menant à une fin de temporisation de libération

L'automate AP_TPL mémorise, via un indicateur, qu'une temporisation de libération a eu lieu alors qu'elle était effectivement attendue. Cet indicateur est utilisé par AP_An. Il y a un automate de ce type par annonce (Fig. 7.41).

### 7.4.3.4 Kontrolle der Sequenz die zur Auflösung einer Verzögerung führt

Der Beweisautomat AP_TPL speichert mithilfe eines Indikators, dass eine Auflösung der Verzögerung stattgefunden hat als diese auch erwartet wurde. Dieser Indikator wird von AP_An benützt. Es gibt einen Automaten dieses Typs pro Vormelder (Abb.7.41).

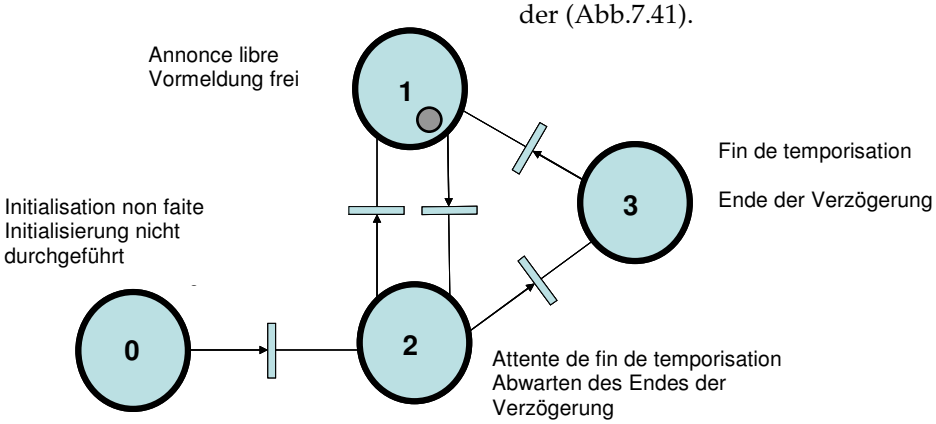



**Figure 7.41** : Automate de preuve de la séquence de réarmement pour une voie et un sens donné (AP_TPL)
**Abbildung 7.41**: Beweisgraph für die Überprüfung der Verzögerungsauflösung - ein Gleis und eine Richtung (AP_TPL)

## 7.4.4 Temps de calcul

La preuve du système complet vérifie l'ensemble des 4 annonces dans la même exploration entraînant un grand nombre d'états système redondants. Les annonces sont indépendantes, la preuve du système peut se découper en 4 preuves. Pour ce faire, il suffit d'interdire lors de l'exploration la modification des entrées terrain intervenant dans les annonces dont la preuve n'est pas en train d'être effectuée. Le nombre d'états explorés et le gain de temps résultant en fonction du nombre d'annonces testées simultanément sont reportés dans le tableau 7.18.

| | *Nombre d'états système* | *Temps nécessaire à la preuve* |
|---|---|---|
| *Une annonce* | 110 | 0h10 min |
| *Une voie et deux annonces* | 206 | 0h30 min |
| *Deux voies* | 10004 | 4h36 min |

**Figure 7.18 :** États systèmes et temps de preuve

Le fait d'utiliser la réunion des preuves de sous ensembles indépendants plutôt que la preuve du système complet apparaît donc clairement comme un aspect à ne pas négliger.

## 7.4.4 Rechenzeit

Der Beweis des Systems überprüft alle vier Vormeldungen auf einmal. Dies führt zu einer großen Anzahl von redundanten Systemzuständen. Da die Vormeldungen unabhängig sind, kann der Beweis in vier Teile geteilt werden. Um dies zu machen reicht es beim Erforschen der Zustände aus, die Änderung der vor Ort Eingänge zu verbieten, die die anderen Vormeldungen betreffen. Die Anzahl der gleichzeitig getesteten Vormeldungen sind in der Tabelle 7.18 zu sehen.

| | *Anzahle der Systemzustände* | *Notwendige Zeit* |
|---|---|---|
| *Eine Vormeldung* | 110 | 0h10 min |
| *Ein Gleis mit zwei Vormeldungen* | 206 | 0h30 min |
| *Zwei Gleise mit vier Vormeldungen* | 10004 | 4h36 min |

**Tabelle 7.18**: Anzahl der System und Prüfzeit

Die Benutzung der zusammengefassten Beweise unabhängiger Teilmengen erscheint im Vergleich zu einem Beweis des kompletten Systems als ein nicht zu vernachlässigender Gesichtspunkt.

# CHAPITRE 8

# Conclusion

## 8.1 Généralité

### 8.1.1 La méthode

Le retour sur les particularités, historiques et techniques, du système ferroviaire et l'identification des attentes du gestionnaire de l'infrastructure et des possibilités nouvelles offertes par le monde universitaire nous ont conduit à définir une méthode de conception et de validation d'applications informatiques critiques.

La méthode présentée dans ce travail :

- a été appliquée sur des fonctionnels applicatifs de postes d'aiguillage en exploitation et est appliquée « en double » sur un poste neufs ;
- permet de traiter les postes neufs et ceux existants modifiés ;
- repose sur des outils permettant d'élaborer les propriétés de sécurité du poste d'aiguillage et de valider formellement celui-ci à leur encontre ;
- propose un compte rendu de preuve avec, le cas échéant, l'identification des états système en défaut et la succession d'événements y conduisant.

Les travaux menés dans ce travail visaient notamment à lever les principales critiques contre les méthodes formelles, à savoir :

- Elles exigent l'écriture des propriétés de sécurité du système ;
- Elles exigent une connaissance détaillée des techniques mathématiques ;
- Elles augmentent des prix de production malgré un gain sur la phase de validation ;
- Elles ne sont pas accompagnées avec les systèmes spécifiés ;
- Elles restent hermétiques pour les utilisateurs (les clients et pas les spécialistes) ;
- Elles ne sont pas utilisables pour les systèmes industriels.

# KAPITEL 8

# Zusammenfassung

## 8.1 Allgemeines

### 8.1.1 Methode

Die Erörterung der historischen und technischen Besonderheiten des Eisenbahnsystems, die Identifikation der Erwartungen des Infrastrukturbetreibers und die neuen Möglichkeiten aus der Forschung und Entwicklung an den Universitäten haben dazu geführt, ein Verfahren zur Konzeption und zur Validierung kritischer IT-Anwendungen zu definieren.

Die in dieser Arbeit vorgestellte Methode:

- wird auf Anwendungsfunktionen bestehender Stellwerke und zusätzlich auf ein neues Stellwerk angewandt.
- erlaubt es, neue und schon vorhandene aber veränderte Stellwerke zu behandeln.
- beruht auf Werkzeugen zur Bestimmung der Sicherheitseigenschaften des Stellwerks und zu deren formalen Validierung.
- schlägt ein Beweisprotokoll vor und erlaubt gegebenenfalls die Aufdeckung der fehlerhaften Systemzustände und die Reihenfolge der Ereignisse, die dazu führen.

Ziel dieser Arbeit ist es, die Hauptkritikpunkte an den formalen Methoden zu widerlegen, das heißt, zu zeigen, dass:

- sie die Formulierung der Sicherheitseigenschaften des Systems erfordern.
- sie die detaillierte Kenntnis der mathematischen Grundlagen erfordern.
- sie nicht die Produktionskosten erhöhen.
- es spezielle Anwendungsprogramme für formale Methoden gibt.
- sie für die Anwender (die Kunden, nicht nur für die Spezialisten) verständlich sind.
- sie für industrielle Systeme verwendbar sind.

## 8.1.2 Travaux

Notre méthode originale est applicable industriellement pour la validation des spécifications fonctionnelles de postes d'aiguillage, permettant ainsi de prouver que les spécifications réalisent bien toutes les exigences attendues et que le produit est correctement conçu d'un point de vue fonctionnel. Cette méthode pourrait être utilisée pour formaliser d'une façon commune pour de nouvelles spécifications de nouveaux postes d'aiguillages. Elle présente les avantages suivants:

- L'élaboration de propriétés de sécurité sera faite par les experts du chemin de fer ayant une connaissance du contexte règlementaire d'exploitation et de maintenance du système ferroviaire non pas par les mathématiciens qui n'ont pas une connaissance approfondie dans les systèmes signalisation ;
- La possibilité d'appliquer la preuve à des fonctionnels existants ou modifiés indépendamment ;
- La possibilité pour les industriels, s'ils veulent interpréter directement les spécifications fonctionnelles qui peuvent leur être confiées, de réduire leurs propres efforts de conception et de validation ;
- L'exploration automatique et systématique des états accessibles permet de vérifier toutes les chronologies d'événement qui ne sont pas inclus dans les essais (économique contraint qui limite le temps d'essai …) ou tous les scénarios pratiquement réalisables (l'occurrence de plusieurs événements sur un délai court…). Ceci dépasse les limites inhérentes aux méthodes de test classiques ;
- L'utilisation des réseaux de Petri permet une bonne lisibilité, une exploitation et compréhension facilitées et un lien avec le passé, une expérience ;
- La méthode est applicable industriellement pour les acteurs du ferroviaire en levant les critiques précédentes : les utilisateurs peuvent n'avoir aucune connaissance des techniques mathématiques, le formalisme permet de réduire les prix de production et de maintenance du logiciel critique. Les outils industriels sont en développement, la preuve d'un poste peut être découpée en plusieurs preuves partielles, rendant le traitement d'un poste complet abordable dans des délais raisonnables.

Cette méthode pourra à terme prendre sa place dans le processus de réalisation des essais avant mise une exploitation à la SNCF sans modification du niveau de sécurité requis pour les essais.

## 8.1.2 Arbeiten

Das neu entwickelte Verfahren ist für die Validierung der funktionellen Spezifikationen der Stellwerke industriell verwendbar. Somit kann bewiesen werden, dass die Spezifikationen allen erwarteten Anforderungen entsprechen und dass das Produkt korrekt konzipiert wurde.

Dieses Verfahren könnte auch der gemeinsamen formalen Gestaltung neuer Spezifikation von neuen Stellwerken dienen. Es bietet folgende Vorteile:

- Die Ausarbeitung der Sicherheitseigenschaften wird von den Bahningenieuren durchgeführt und nicht von Mathematikern, die keine grundlegenden Kenntnisse der Signaltechnik haben.
- Die Möglichkeit, die Beweisführung auch auf bestehende oder auf einzeln veränderte Funktionen anzuwenden.
- Die Möglichkeit für die Industrie, den Konzeptions- und Validierungsaufwand zu reduzieren, falls sie die ihnen übertragenen funktionellen Spezifikationen direkt interpretieren wollen.
- Durch die automatische und systematische Untersuchung der erreichbaren Zustände können alle zeitlichen Abläufe der Ereignisse, die normalerweise nicht untersucht werden (wirtschaftliche Randbedingungen, die die Versuchszeit einschränken…) und alle praktisch realisierbaren Szenarien (das Zusammentreffen zweier Ereignisse in kürzester Zeit…) überprüft werden. Dies geht über die Grenzen der klassischen Testverfahren hinaus.
- Die Anwendung der Petrinetze erleichtert die Lesbarkeit, den Betrieb und das Verständnis der Systeme.
- Das Verfahren lässt sich von der Eisenbahnindustrie industriell anwenden, falls die zuvor genannten Kritikpunkte beseitigt werden: die Anwender müssen die mathematischen Techniken nicht kennen, der Formalismus erlaubt es, die Produktions- und die Wartungskosten der kritischen Software zu reduzieren, die industriellen Werkzeuge sind in Entwicklung und die Beweisführung für ein Stellwerk kann in Teilbeweisführungen untergliedert werden, wodurch die Behandlung des kompletten Stellwerks in einer angemessenen Zeit durchführbar ist.

Dieses Verfahren wird bei der SNCF einen Platz im Versuchsablauf vor der Inbetriebnahme einnehmen und zwar ohne Änderung des bei den Versuchen erforderlichen Sicherheitsniveaus.

### 8.1.3 Discussion

Pour l'instant, les méthodes formelles ou semi formelles basées sur la modélisation sont généralement considérées, à tort d'ailleurs, comme coûteuses en ressources (humaines et matérielles) et souvent réservées aux logiciels les plus critiques et nécessitant avant tout de complètes compétences métier en mathématiques.

Leur amélioration et l'élargissement de leurs champs d'application pratique sont la motivation de nombreuses recherches scientifiques en informatique.

Avant tout, la maîtrise de ces démarches de modélisation, et leur rentabilisation devront s'appuyer sur une évolution organisationnelle et culturelle des entreprises qui banalisera l'utilisation des modèles et l'exploitation des traitements formels dont ils pourront être l'objet.

Cependant, n'oublions jamais qu'un modèle est une description partielle d'un système selon un point de vue particulier se limitant à représenter seulement certains aspects pertinents de l'univers du système que l'on modélise.

Il faut donc toujours garder à l'esprit qu'un modèle traduira toujours un ou plusieurs point(s) de vue partiel(s) sur une application logicielle, et même si tous les aspects comportementaux intrinsèques d'une application logicielle peuvent être couverts par un langage de modélisation, il y a peu de chance que l'ensemble des combinatoires des interactions de cette application avec les différentes composantes de son environnement puissent être couverts par un seul modèle.

Il est donc important d'apprendre à qualifier précisément le périmètre de couverture et d'interprétation d'un modèle et des résultats associés à travers une maîtrise et une formalisation précises des postulats initiaux qu'il suppose (hypothèses de comportement, d'interaction avec l'environnement...) et des domaines de définition, de variation et de validation de la totalité des entrants souhaités ou non souhaités.

Pour nous, l'idéal consiste à exécuter directement le modèle formel qui a servi à produire des preuves et à démontrer des propriétés de manière absolue, sans passer par un processus de génération de code algorithmique.

### 8.1.3 Diskussion

Zurzeit werden die auf Modellierung beruhenden formalen oder semi-formalen Methoden im Allgemeinen (im Übrigen zu Unrecht) als ressourcenaufwändig bezeichnet (menschliche und physikalische Ressourcen) und oft werden sie nur für die kritischste Software verwendet.

Die Verbesserung dieser Methoden und die Erweiterung ihrer praktischen Anwendungsbereiche sind die Motivation zahlreicher wissenschaftlicher Forschungsarbeiten im IT-Bereich.

Die Beherrschung dieser Modellierungsverfahren und deren Wirtschaftlichkeit müssen sich vor allem auf eine organisatorische und kulturelle Entwicklung der Unternehmen stützen. Dadurch werden die Verwendung von Modellen und deren formale Bearbeitung kommerzialisiert.

Es darf jedoch niemals vergessen werden, dass ein Modell eine Teilbeschreibung des Systems ist, unter einem ganz besonderen Gesichtspunkt, wodurch lediglich bestimmte zutreffende Aspekte des modellierten Systemsoftwarebereichs dargestellt werden.

Weiterhin muss bedacht werden, dass ein Modell immer einen oder mehrere Teilstandpunkte einer Softwareanwendung darstellt, und selbst wenn alle Verhaltensaspekte einer Softwareanwendung von einer Modellierungssprache abgedeckt werden können, so ist es unwahrscheinlich, dass alle Kombinationen der Wechselwirkungen dieser Anwendung mit den verschiedenen Komponenten des Umfeldes von einem einzigen Modell abgedeckt werden.

Es ist also wichtig, den Deckungs- und den Interpretationsgrad eines Modells und der entsprechenden Ergebnisse genau zu qualifizieren, dank einer guten Kenntnis, Beherrschung und präziser Formalisierung der ursprünglichen Anforderungen (Annahmen über das Verhalten, über die Wechselwirkung mit dem Umfeld...) und der Definitions-, Variations-, und der Validierungsbereiche aller gewünschten und unerwünschten Eingaben.

Die ideale Vorgehensweise besteht darin, das formale Modell zur kompletten Beweisführung der Eigenschaften direkt und ohne den Umweg über eine Codeerzeugung (und seine Kompilierung) auszuführen.

Dès lors que ces précautions et ces réserves sont acquises, il est clair que l'utilisation des méthodes formelles reste la manière privilégiée d'établir une garantie sur le comportement d'une application logicielle : on a bien alors la possibilité de démontrer qu'une situation redoutée ne sera jamais atteinte, ou qu'une propriété sera toujours vérifiée[78], cela quels que soient les aléas d'interaction de l'application avec son environnement.

Autrement dit, on a la possibilité de démontrer le «jamais» ou le «toujours», et pas seulement le «peut-être» modulé par un chiffre de probabilité ou de gravité...

Enfin, le référentiel méthodologique et technique de l'Ingénierie de Modèles permettra de moduler et de maîtriser plus précisément encore le périmètre d'application et d'interprétation des résultats issus des méthodes formelles, en fonction des couches physiques ou logicielles constituant l'environnement de l'application. L'Ingénierie de Modèles nous permettra alors d'affronter les autres enjeux liés à la pérennité des applications et des statuts de sécurité établis sur cette application à un moment donné de son cycle de vie :

- preuve, démonstration des propriétés de sécurité à un instant donné ;
- indépendance de ces statuts de preuve vis-à-vis des couches logicielles et physiques d'implémentation de l'application ;
- pérennité de ces propriétés de sécurité dans le temps, en fonction des phénomènes d'évolutions ou d'obsolescences susceptibles de caractériser l'environnement immédiat et moins immédiat de l'application ;
- minimisation des coûts consacrés au maintien de ces statuts de sécurité dans le temps.

In dem Moment, in dem diese Vorsichtsmaßnahmen und Vorbehalte berücksichtigt werden, ist die Anwendung des formalen Verfahrens die bevorzugte Art, ein bestimmtes Verhalten eines Programms zu gewährleisten. Es besteht nämlich die Möglichkeit zu beweisen, dass ein gefährliches Ereignis niemals eintreten wird[79] oder dass eine Eigenschaft immer wahr ist, wie auch immer die zufallsbedingten Wechselwirkungen der Anwendung mit ihrem Umfeld sein werden.

Mit anderen Worten, es besteht die Möglichkeit, das „nie“ oder das „immer“, nicht nur das modulierte „vielleicht“ mit einer Wahrscheinlichkeitsangabe oder einem Schadensausmaß zu beweisen.

Die methodischen und die technischen Referenzen des für die Modelle zuständigen Ingenieurwesens werden es ermöglichen, den Umfang der Anwendung und der Interpretation der Ergebnisse der formalen Verfahren noch präziser zu modulieren und zu beherrschen, in Abhängigkeit von den physikalischen oder programmtechnischen Fällen der Anwendungsumgebungen. Das für die Modellierung zuständige Ingenieurwesen ermöglicht es, sich den anderen Herausforderungen in Verbindung mit dem Langzeitverhalten und dem Sicherheitszustand zu stellen:

- vollständige Beweisführung der Sicherheitseigenschaften zu einem bestimmten Zeitpunkt
- Unabhängigkeit dieser Beweisführungen von der softwaremäßigen und physikalischen Implementierung der Anwendung
- zeitliche Stabilität der Sicherheitseigenschaften, in Abhängigkeit der Entwicklungen und der Alterungserscheinungen, die das direkte und das weniger direkte Umfeld der Anwendung charakterisieren können
- Minimierung der Kosten für die Langzeitaufrechterhaltung dieser Sicherheitszustände.

[78] Déjà en 1898 monsieur Ddescubes parlait de „sécurité absolue“ [Descubes, 1898]

[79] Schon in 1898,spach Herr Descubes von „absolute Sicherheit, [Descubes, 1898]

### 8.1.4 Résultats

Les travaux ont mené à une méthode originale applicable industriellement pour la réalisation et la validation des spécifications fonctionnelles exécutables de postes d'aiguillage. Dans ce contexte la possibilité existe de prouver automatiquement que les spécifications réalisent bien toutes les exigences attendues et que le produit est correctement conçu. Le langage AEFD défini pourrait être utilisé pour formaliser d'une façon commune (entre pays) des spécifications de nouveaux postes d'aiguillages ou produits assimilés.

Nous montrerons que la méthode est applicable industriellement en levant les critiques précédentes : les utilisateurs peuvent n'avoir aucune connaissance des techniques mathématiques, le formalisme permet de réduire des prix de production et de maintenance du logiciel critique, les outils industriels sont en cours de développement, la preuve d'un poste peut être découpée en plusieurs preuves partielles, rendant le traitement d'un poste complet abordable dans des délais raisonnables…

## 8.2 Du point de vue universitaire

Nos travaux peuvent jeter un pont entre les mondes universitaires et industriels dans l'intérêt mutuel des deux mondes et de la collectivité.

Le premier dispose en effet de nombre de théorie, méthodes, approches… relatives Réseaux de Pétri et plus particulièrement aux « automates à nombre d'états fini » dans leurs diverses formulations.

Le second doit dès à présent concevoir, maintenir et faire évoluer (vivre) de nombreux systèmes critiques de sécurité, systèmes informatiques où toutes les fonctionnalités sont spécifiables par les automates à états finis.

Notre approche montre qu'il est possible de réaliser physiquement avec les contraintes inhérentes au développement de système SIL4, une machine cible se comportant exactement comme un automate à nombre d'états finis. Le «modèle» spécifiant l'applicatif fonctionnel passe du statut d'«abstraction de la réalité» et de «spécification à réaliser» à celui de «description réelle» ou de «spécification exécutée».

### 8.1.4 Ergebnisse

Die vorliegende Arbeit führt zu einem neuartigen, industriell anwendbaren Verfahren zur Realisierung und Validierung der ausführbaren funktionellen Spezifikationen von Stellwerken. Es besteht die Möglichkeit, automatisch zu beweisen (die manuelle Validierung zur Abdeckung der Verschlusslücken bei der Formulierung der AP ist gewollt), dass die Spezifikationen allen Anforderungen entsprechen und dass das Produkt korrekt konzipiert ist. Die definierte AEFD-Sprache könnte zur gemeinsamen (Kooperation verschiedener Länder) formalen Gestaltung von Spezifikationen für neue Stellwerke oder ähnliche Produkte dienen.

Das Verfahren lässt sich industriell anwenden, falls die zuvor genannten Kritikpunkte beseitigt werden: die Anwender müssen die mathematischen Techniken nicht kennen, der Formalismus erlaubt es, die Produktions- und die Wartungskosten der kritischen Software zu reduzieren, die industriellen Werkzeuge sind in der Entwicklung und die Beweisführung für ein Stellwerk kann in Teilbeweisführungen untergliedert werden, wodurch die Behandlung des kompletten Stellwerkes in einer vernünftigen Zeit durchführbar ist…

## 8.2 Entwicklungen an Universitäten

Diese Arbeit kann eine Brücke bauen zwischen der Welt der Universität und der Welt der Industrie, im gegenseitigen Interesse und im Interesse der Allgemeinheit.

Die Universitäten verfügen über zahlreiche Theorien, Verfahren, Vorgehensweisen in Verbindung mit Petrinetzen und insbesondere bezüglich der „endlichen Automaten“ in unterschiedlichen Formulierungen.

Die Industrie muss zahlreiche kritische Sicherungssysteme und IT-Systeme konzipieren, warten und weiterentwickeln; dabei können alle Funktionalitäten mit endlichen Automaten spezifiziert werden.

Die Vorgehensweise in dieser Arbeit zeigt die Möglichkeit der physikalischen Realisierung einer Zielmaschine, die sich genauso verhält wie ein endliche Automat, unter Berücksichtigung der mit der Entwicklung von SIL4-Systemen einhergehenden Randbedingungen. Der Zustand „Abstraktion der Realität“ des „Modells“, das die funktionelle Anwendung spezifiziert, entwickelt sich weiter zu dem Zustand einer «zu realisierenden Spezifikation» und danach zu dem Zustand einer „wirklichen Beschreibung“ oder einer „ausführbaren Spezifikation“.

J'espère que cette voie pourra permettre de diffuser dans le monde industriel les résultats des travaux des chercheurs de tous pays sur les réseaux de Petri en général, sur les automates à nombre d'états fini en particulier. Dans une certaine mesure c'est redonner vie à la démarche initiée dans les années 1970 autour du GRAFCET mais avec une autre ouverture.

## 8.3 Du point de vue de l'industrie ferroviaire

Il existe sur le marché des outils de génération de code, certains sont associés à une analyse de risques, qui visent à obtenir un code de meilleure qualité grâce à une démarche formelle ou semi formelle.

Mais celle-ci est alors efficace que dans la mesure où le modèle (abstraction de la réalité) contient déjà toutes les informations fonctionnelles nécessaires à une élaboration du logiciel exécuté sans transformation algorithmique.

De manière simplifiée, le modèle devient «la spécification réalisable» (ou à réaliser) afin d'effectuer au plus tôt les scenarii de tests ou les analyses de structures au niveau du modèle. Cela revient à reconnaître que les méthodes actuelles de développement des logiciels génèrent des erreurs de spécification, des fautes de conception ou des fautes d'implémentation (haut niveau) qu'il convient d'identifier au plus tôt pour un traitement peu onéreux.

Raisonner sur les modèles amont est incontestablement un progrès, seulement si les concepteurs prennent en compte tous les éléments de contexte ferroviaire et d'environnement du futur système. Néanmoins, il reste toujours le pas conséquent (risqué) de l'implémentation pratique (physique) sur les calculateurs cibles. Nous avons montré que notre approche combinée «matériel et logiciel» permet de traiter ces risques de manière efficace, du point de vue pratique et économique. De manière simplifiée, le modèle devient «la spécification exécutable» afin d'effectuer au plus tôt la preuve formelle du fonctionnel applicatif ainsi défini et qui sera exécuté de manière déterministe.

Es besteht die Hoffnung, die weltweiten Forschungsergebnisse bezüglich Petrinetzen ganz allgemein und im Zusammenhang mit endlichen Automaten im Besonderen auch in der Industrie bekannt zu machen. Die Ergebnisse dieser Arbeit sind in gewisser Weise die Wiederbelebung einer Initiative aus den siebziger Jahren, in Verbindung mit dem GRAFCET, aber in eine andere Richtung.

## 8.3 Aus Sicht der Bahnindustrie

Auf dem Markt existieren Anwendungsprogramme zur Codeerzeugung. Einige davon sind an eine Risikoanalyse gekoppelt mit dem Ziel einen Code besserer Qualität zu erreichen, dank einer formalen bzw. einer semi-formalen Vorgehensweise.

Dies ist möglich, wenn das Modell (Abstraktion der Wirklichkeit) bereits alle Informationen über die Software ohne algorithmische Transformation beinhaltet.

Das Modell vereinfacht ausgedrückt, wird zu einer „realisierbaren (oder zu realisierenden) Spezifikation", um die Testszenarien oder die Strukturanalysen auf der Ebene des Modells so früh wie möglich durchzuführen. Die heutigen Softwareentwicklungsverfahren bringen Spezifikations-, Konzeptions- oder Implementierungsfehler (auf hoher Ebene) mit sich, die so schnell wie möglich identifiziert werden müssen, damit deren Behandlung nicht zu teuer wird.

Die Tatsache, dass man vorab mit Modellen arbeitet, ist unwiderlegbar ein Fortschritt, aber nur dann, wenn die Entwickler auch alle Elemente des Bahn- und des künftigen Systemumfeldes berücksichtigen. Es bleibt jedoch immer noch der weitreichende und riskante Schritt der praktischen (physikalischen) Implementierung auf den Zielrechnern. Es ist bewiesen, dass die kombinierte Vorgehensweise „Hardware und Software" es erlaubt, diese Risiken effizient, praktisch und wirtschaftlich auszuräumen. Vereinfacht ausgedrückt wird das Modell zur „ausführbaren Spezifikation", um so früh wie möglich den formalen Beweis der so definierten Anwendungsfunktionen, die deterministisch ausgeführt werden, zu erbringen.

Le 7 décembre 1835 la locomotive « Adler » roulait à jusqu'à 65 km/h de Nuremberg à Fürth initiant ainsi la dynamique ferroviaire en Allemagne. La sécurité reposait alors essentiellement sur l'expérience du chauffeur et le bon oeil du mécanicien. A présent, en France comme en Allemagne, les circulations dépassent les 300 km/h, visent le 350 km/h en service commercial, avec des conditions de débit toujours plus exigeantes. La sécurité repose maintenant largement sur des ordinateurs, tant à bord qu'au sol.

Les fonctions de sécurité étant réalisées aussi en grande partie dans le logiciel, les législateurs, nationaux et européens, exigent naturellement la garantie d'un haut niveau de sécurité (la qualité et fiabilité du logiciel) au moyen de normes. La norme européenne applicable actuellement dans le domaine ferroviaire, l'EN 50128, présente un processus, des procédés et des principes au bout desquels la qualité du logiciel doit être convenable en regard des exigences de sécurité.

L'EN 50128 est une norme de processus et définit un processus de développement de logiciel (essentiellement le logiciel de base). La norme La norme exige une démarche Top Down dans la gestion des projets, une conception modulaire, une vérification après chaque pas de développement, une documentation traçable et vérifiable ainsi que des procédures de tests. C'est une démarche classique d'assurance qualité du processus de conception et de génération des logiciels.

## 8.3.1 Essais des logiciels sur la machine cible

En raison de la longévité de l'exploitation des systèmes contribuant à la circulation des trains, 30 ans *a minima*, beaucoup d'applications sont encore programmées en assembleur, Pascal, C. De ce fait, de nombreuses demandent d'évolution apparaissent, comme par exemple des déplacements de spécification, des obsolescences matérielles qui doivent être testées manuellement bien au delà des modifications effectuées, ce qui représente un facteur de frais considérable, sans qu'aucune garantie totale ne puisse être apportée.

L'architecture proposée avec son logiciel de base réutilisable en l'état, avec pour seul paramétrage le fonctionnel lié aux besoins du site, et peu dépendant de la plateforme matérielle présente rapidement un fort intérêt.

Am 7. Dezember 1835 fuhr die Lokomotive „Adler" bereits mit 65 km/h von Nürnberg nach Fürth. Sie hat die Bahndynamik in Deutschland ausgelöst. Damals stützte sich die Sicherheit im Wesentlichen auf die Erfahrung des Heizers und auf die guten Augen des Lokomotivführers. Gegenwärtig wird in Deutschland wie auch in Frankreich mit einer Geschwindigkeit von mehr als 300 km/h gefahren. Zusätzlich werden die Verkehrsflussbedingungen immer strenger. Heute beruht die Sicherheit im Wesentlichen auf Computern, die an Bord der Fahrzeuge oder in der Infrastruktur integriert sind.

Da die Sicherheitsfunktionen größtenteils auch in der Software stecken, fordern die nationalen und europäischen Gesetzgeber selbstverständlich die Gewährleistung eines hohen Sicherheitsniveaus (Qualität und Zuverlässigkeit der Software) auf der Grundlage von Normen. Die zurzeit im Bahnbereich gültige europäische Norm (EN 50128) beinhaltet einen Prozess, Prozeduren und Prinzipien, wonach die Qualität der Software im Hinblick auf die Sicherheitsanforderungen angemessen sein soll.

Die EN 50128 ist eine Prozessnorm und definiert ein Verfahren für die Programmentwicklung (für Grundsoftware). Sie fordert eine Top-Dow-Vorgehensweise beim Management der Projekte, einen modularen Aufbau, eine schrittweise Überprüfung der Entwicklung, eine nachvollziehbare und überprüfbare Dokumentation sowie Testverfahren. Es handelt sich dabei um eine klassische Vorgehensweise der Qualitätssicherung für die Konzeption und die Erzeugung von Programmen.

## 8.3.1 Softwaretest auf dem Zielrechner

Aufgrund der Langlebigkeit der Zugsteuerungssyteme (mindestens 30 Jahre) sind noch viele Anwendungen in Assembler, Pascal oder C programmiert. Aus diesem Grund besteht eine große Nachfrage nach Entwicklung: Spezifikationsänderungen, Hardwareveralterung, usw. müssen manuell getestet werden, weit über die durchgeführten Änderungen hinaus, was einen beachtlichen Kostenfaktor darstellt, ohne dass eine Gesamtgewährleistung erreicht werden kann.

Die in dieser Arbeit vorgestellte Architektur mit dem ohne Änderungen wiederverwendbaren Grundprogramm, das als einzigen Parameter die den Bedürfnissen des Standards entsprechenden Funktionen hat und das sehr wenig von der Hardwareplattform abhängig ist, wird so sehr interessant.

### 8.3.2 Spécification interprétable

C'est en grande partie sur ce constat que reposent les méthodes modernes comme le développement à base de modèles. C'est le cas d'ateliers (par exemple SCADE) qui se fondent sur le langage semi formel UML (Unified Modelling) pour la conception complète de l'application. C'est un progrès, mais le code final reste uniquement « certifié ». Les « modèles » peuvent être exportés aussi complètement sur machine de test, analysés finement et faire l'objet d'un contrôle de l'aptitude à l'emploi. Les «modèles» produits sont alors réputés '*correct by construction*' parce qu'ils sont contrôlés automatiquement par des outils de contrôle de code (division par zéro, boucles infinie, structures…).

### 8.3.3 Essais et validation formelle

Puisque le modèle représente la description complète de l'application, tous les tests peuvent être effectués excepté le test d'intégration ainsi que les essais réels sur machine cible. La réalisation d'une validation/preuve formelle permet le contrôle du modèle sur ses propriétés avec des méthodes mathématiques.

Contrairement à des tests avec un nombre limité de cas de test, une preuve de correction mathématique des spécifications est réalisable sur l'ensemble du domaine des possibles des entrées.

Là encore, il reste toujours le pas de l'implémentation pratique (physique) sur les calculateurs cibles, sans transformation de nature à réduire le champ de la validité des travaux de validation réalisés.

### 8.3.4 Génération de codes et compilateur certifiés

Décrire l'application en UML permet une génération automatique «certifiée» de codes et de documents et est acceptée actuellement pour l'EN50128 SIL3\4. Le procédé de certification est alors simplifié puisque la preuve apporte normalement une équivalence avec les essais réalisés antérieurement, au moins à haut niveau.

### 8.3.5 Le modèle et l'implémentation

Le code de Source final est produit à partir des modèles pour la machine cible (options de compilations pour être intégré sur la machine cible). Pour réduire le risque d'erreur certains ateliers utilisent un compilateur, lequel contrôle la traduction correcte des constructions linguistiques et paramétrages physiques.

### 8.3.2 Interpretierbare Spezifizierung

Moderne Verfahren, wie die Entwicklung auf der Grundlage von Modellen, beruhen im Wesentlichen auf dieser Feststellung. Das gilt auch für die Entwicklungsumgebungen (z.B SCADE), die auf der semi-formalen UML-Sprache (Unified Modelling) für die vollständige Konzeption der Anwendung beruhen. Dies ist bereits ein Fortschritt, aber der Endcode ist lediglich „zertifiziert". Die „Modelle" können vollständig auf eine Testmaschine exportiert und detailliert analysiert werden, bevor sie einer Eignungskontrolle unterzogen werden. Die so entstehenden „Modelle" gelten als „*korrekt durch Beweis*", weil sie von Codekontrollprogrammen (Teilung durch null, unendliche Schleifen, Strukturen, usw.) automatisch kontrolliert werden.

### 8.3.3 Test und formale Überprüfung

Da das Modell die Anwendung vollständig beschreibt, können alle Tests durchgeführt werden, mit Ausnahme des Integrationstests und der tatsächlichen Versuche auf der Zielmaschine. Die Durchführung einer Validierung und eines formalen Beweises erlaubt es, die Eigenschaften des Modells auf der Grundlage von mathematischen Verfahren zu kontrollieren.

In Gegensatz zu Tests mit einer beschränkten Anzahl von Testfällen, kann der Beweis der mathematischen Korrektheit der Spezifikationen erbracht werden und zwar für den gesamten Bereich der möglichen Eingangsgrößen.

Auch hier verbleibt noch der Schritt der praktischen (physikalischen) Implementierung auf den Zielrechnern, ohne Umwandlung, die den Gültigkeitsbereich der durchgeführten Validierung reduzieren würde.

### 8.3.4 Zertifizierte Codeerzeugung und Überprüfungscompiler

Die UML Beschreibung der Anwendung erlaubt es, automatisch „zertifizierte" Codes und Dokumente zu erzeugen und wird von der EN 50128 SIL3\4 akzeptiert. Das Zulassungsverfahren ist in diesem Fall vereinfacht, zumal die Beweisführung normalerweise gleichgestellt wird mit den früheren Versuchen, zumindest auf höchster Ebene.

### 8.3.5 Modell und Implementierung

Der endgültige Quellcode wird gemäß der Modelle für die Zielmaschine (Kompilationsoptionen, zur Integration in die Zielmaschine) erzeugt. Um das Fehlerrisiko einzugrenzen, verwenden bestimmte Entwicklungsumgebungen einen Überprüfungscompiler, der die korrekte Übersetzung der linguistischen Konstruktionen und der physikalischen Parametrisierungen kontrolliert.

Ce traducteur est certifié pour des machines cibles identifiées afin de couvrir sans essais supplémentaires la fin de la démarche de conception de la spécification au code compilé exécutable.

Quelle confiance apporter à ces outils «certifiés», cible par cible en pratique ? Quelle confiance apportée à ces contrôles de codes ? A l'évidence, dans la mesure où l'application le permet, il sera toujours préférable de miser sur l'absence de toute transformation entre le modèle prouvé et celui exécuté sur la machine cible. La valeur ajoutée des méthodes formelles, incluant l'application et son environnement, couvre ainsi tout le cycle de développement. C'est l'avantage majeur de « la spécification exécutable ». Il n'est alors plus nécessaire de faire des analyses de taux de couverture des tests effectués, demandé par la norme EN 50128, la validation étant totale et valide sur la machine cible.

La conception *ex nihilo* de logiciels critiques est relativement peu fréquente, généralement il s'agit de modifications, d'extensions ou d'adaptation de logiciels existants. Afin d'envisager l'avenir il est donc nécessaire de prendre en compte ce besoin en laissant la main aux gens de métier pour faire évoluer les fonctionnalités du système sans remettre en cause les fonctions antérieurement développées. Les démarches de conception et validation formelles se heurtent ici à des difficultés. L'approche présentée repose une bibliothèque évolutive de *graphes génériques instanciables* qui, si l'on respecte les règles de conception édictée, ne remettent pas en cause le caractère déterministe et prouvable d'un nouveau fonctionnel. L'application d'une méthode de « validation formelle » permet alors de prouver l'ensemble, code précédent et nouveau code, sans distinction et en temps masqué.

Ce travail avait pour objet de proposer une méthode formelle qui permette, outre les gains économiques et de sécurité attendus, d'assurer :

- les conditions de réussite de l'application de méthodes formelles dans le domaine particulier de l'industrie ferroviaire ;
- le maintien de la maîtrise fonctionnelle et technique des installations de sécurité par les acteurs du ferroviaire.

Dieser Übersetzer ist für einige Zielmaschinen zertifiziert und erlaubt es, ohne zusätzliche Versuche am Ende der Konzeption der Spezifikation des ausführbaren Codes auszukommen.

Welches Vertrauen kann man in diese „zertifizierten" Programme in der Praxis setzen? Welches Vertrauen kann man den Codekontrollen schenken? Jegliche Umwandlung zwischen dem bewiesenen Modell und dem auf der Zielmaschine ausgeführten, ist, falls die Anwendung dies erlaubt, zu vermeiden. Der Mehrwert des formalen Verfahrens, einschließlich der Anwendung und des Umfeldes, deckt somit den gesamten Entwicklungszyklus ab. Das ist der Hauptvorteil der „ausführbaren Spezifikation". Es besteht dann keine Notwendigkeit mehr den Deckungsgrad der gemäß EN 50128 durchgeführten Tests zu analysieren, da eine vollständige Validierung auf der Zielmaschine durchgeführt wurde.

Die *ex nihilo* Konzeption kritischer Software ist relativ selten. Im Allgemeinen handelt es sich um Änderungen, Erweiterungen oder Anpassungen von bestehender Software. Für die Zukunft muss dieser Bedarf also berücksichtigt werden. Man muss jedoch den Spezialisten genügend Spielraum gewähren, um die Systemfunktionen weiterzuentwickeln, ohne früher entwickelte Funktionen in Frage zu stellen. Die Vorgehensweisen bei der Konzeption und bei der formalen Validierung stoßen hier auf Schwierigkeiten. Die präsentierte Vorgehensweise beruht auf einer entwicklungsfähigen Bibliothek von generischen (allgemeingültigen) Graphen, die, falls die Konzeptionsregeln eingehalten werden, den deterministischen und nachweisbaren Charakter von neuen Funktionen nicht in Frage stellen. Die Anwendung eines Verfahrens zur „formalen Validierung" erlaubt es, den alten und den neuen Code gemeinsam, ohne Unterscheidung, zeitsparend zu prüfen.

Diese Arbeit hat zum Ziel, eine formale Methode vorzuschlagen, die es erlaubt, neben dem zu erwartenden wirtschaftlichen und sicherheitstechnischen Gewinn folgendes zu gewährleisten:

- eine erfolgreiche Anwendung von formalen Methoden im besonderen Bereich der Eisenbahnindustrie
- die Aufrechterhaltung der funktionellen und technischen Beherrschung der Sicherheitsvorrichtungen durch die Eisenbahnbediensteten.

J'espère avoir montré que ces objectifs peuvent être atteints pour peu que l'on arrête de faire de « l'informatique » et de la « sûreté de fonctionnement » comme souvent aujourd'hui, sans prise en compte suffisante des aspects métier et du système ferroviaire.

Il est important pour les gestionnaires d'infrastructure de spécifier les systèmes informatisés de manière à imposer, outre les fonctions attendues (CdCF), les deux interfaces :

- fonctionnelle en langage AEFD ;
- technique en interface relais.

La spécification des fonctions du système nouveau en langage AEFD permet d'obtenir les assurances que :

- les spécifications sont correctes dans le contexte d'usage ;
- les postulats et les propriétés de sécurité, inconnues dans le détail par les industriels, sont bien remplis ;
- que le gestionnaire d'infrastructure a les moyens de maîtriser le produit tout au long de sa durée de vie, y compris en cas de disparition de l'industriel.

Tout ceci va dans le sens d'une réduction des coûts de développement et de maintenance, sans réduction du niveau de sécurité et avec une prise en compte de la maintenabilité à la conception. Ainsi, outre l'apport initial des méthodes formelles en terme de sécurité, la démarche proposée contribue largement au bilan économique de systèmes informatiques.

Ich hoffe gezeigt zu haben, dass diese Zielsetzungen erreicht werden können, sofern man die „Informatik" und die „Funktionssicherheit" nicht so angeht wie heutzutage oft üblich, ohne ausreichende Berücksichtigung der Gesichtspunkte des Fachgebiets.

Es ist wichtig, dass die Infrastrukturbetreiber die IT-Systeme spezifizieren, um zusätzlich zu den erwünschten Funktionen auch zwei Schnittstellen zu verlangen:

- funktionelle Schnittstelle in AEFD-Sprache
- technische Schnittstelle mit Relais.

Die Spezifizierung der Funktionen neuer Systeme, in AEFD-Sprache durchgeführt, erlaubt es, sich sicher zu sein, dass:

- die Spezifizierungen in dem betroffenen Umfeld korrekt sind.
- die Anforderungen und Sicherheitseigenschaften, die den Herstellern nicht bis ins Detail bekannt sind, erfüllt sind.
- es dem Infrastrukturbetreiber möglich ist, die Anlage über die gesamte Lebenszeit hinweg zu beherrschen, selbst wenn der Hersteller vom Markt verschwindet.

Das alles kann die Entwicklungskosten und Instandhaltungskosten reduzieren, ohne das Sicherheitsniveau zu verringern und unter Einbeziehung der Instandhaltbarkeit von Anfang der Konzeption an. Auf diese Weise bringt die hier vorgestellte Vorgehensweise nicht nur, wie ursprünglich gedacht, Vorteile in Bezug auf die Sicherheit mit sich, sondern verbessert auch sehr die wirtschaftliche Bilanz von IT-Systemen.

# ANNEXES

# ANHANG

# ANNEXE A

# Signalisation française

# ANHANG A

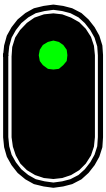
# Französische Signaltechnik

# A.1 Les signaux de cantonnement

# A.1 Blocksignale

[Retiveau, 1987] [Gernigon, 1998] [SNCF, 1941] [SNCF, 1945] und http://www.stellwerke.de/

*État « Ouvert »* **„Offener" Zustand**



*Voie libre :* VL

*indique que la circulation en marche normale est autorisée*

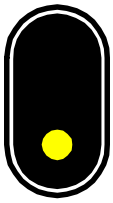
Freie Strecke: VL

Zeigt an, dass eine normale Vorbeifahrt erlaubt ist

*Bemerkung:*

*In Frankreich werden die Lichtbilder der Vor- und der Hauptsignale mithilfe desselben Signals angezeigt.*

*État «Fermé» : indications d'annonce* **„Geschlossener" Zustand: Vorsignale**



*Avertissement :* A

*commande d'être en mesure de s'arrêter avant le signal suivant*

Vorsignal: A

Befehl, in der Lage zu sein, vor dem nächsten Signal anzuhalten. Entspricht dem deutschen Vorsignal „Halt erwarten"

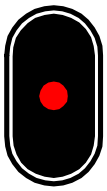
*Jaune clignotant :* (A)

*commande d'être en mesure de s'arrêter avant le signal d'arrêt annoncé à distance réduite par l'avertissement suivant*

Blinkendes Gelb: (A)

Befehl in der Lage zu sein, vor dem Hauptsignal anzuhalten, welches vom nächsten Vorsignal mit einem zu kurzen Abstand angekündigt wird

*État « Fermé » : indications d'exécution* **„Geschlossener" Zustand: Hauptsignale**



*Sémaphore :* S

*commande l'arrêt avant le signal, la plaque d'identification définit les possibilités de franchissement*

Semaphor (Blockhalt): S

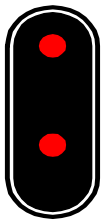
Befehl, vor dem Signal anzuhalten. Eventuelle Schilder geben an, ob das Signal auf Sicht überfahren werden kann.

*Rouge clignotant :* (S)

*commande le franchissement du signal à 15km/h puis la marche à vue à 30 km/h après franchissement jusqu'au signal suivant*

Blinkendes rot: (S)

Weiterfahrt auf Sicht mit 30 km/h bis zum nächsten Signal; das blinkende Rot darf allerdings nur mit 15 km/h überfahren werden.



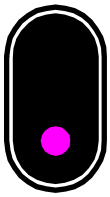
*Carré :* C

*commande l'arrêt avant le signal,*

*n'est franchissable que sur ordre écrit*

Rotes Hauptsignal: C

Absolutes Haltsignal. Darf nur nach schriftlicher Anweisung überfahren werden.



*Carré violet :* Cv

*commande l'arrêt avant le signal sur voie de service, n'est franchissable que sur ordre écrit*

Violettes Hauptsignal: Cv

Absoluter Halt vor dem Signal auf Rangier- und Dienstgleisen

Les carrés (rouge et violet) sont destinés à la protection des nez à nez, des prises en écharpe et des entrebâillements. Les sémaphores et rouges clignotants sont destinés à la protection des rattrapages, dans certains cas, en fonction de la plaque qu'il porte l'agent de conduite peut les franchir de lui même sans avis d'un opérateur sédentaire.

Die Hauptsignale (rot und violett) sind für den Schutz vor Frontalzusammenstößen, Flankenfahrten und Zungenklaffen bestimmt. Das Semaphor und das blinkende Rot schützen vor dem Auffahren; in bestimmten Fällen, je nachdem welches Schild unter dem Signal angebracht ist, kann der Lokomotivführer alleine, ohne Anweisung eines Bahnangestellten im Stellwerk, das Signal überfahren.

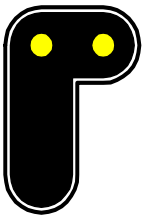
## A.2 Les signaux de limitation de vitesse

Les limitations de vitesse pour les voies déviées des appareils de voie sont présentées selon la position de l'appareil. Elles sont signalisées par des indications lumineuses à 30 et à 60km/h et par caissons lumineux mobiles pour les autre taux de vitesse. Les limitations de vitesse pour les zones fixes sont signalisées par caissons lumineux fixes ou pancartes réflectorisées fixes.

## A.2 Langsamfahrsignale

Die Geschwindigkeitsbegrenzungen für die abzweigenden Gleise der Weichen werden je nach Stellung der Weiche angezeigt. Sie werden durch leuchtende Hinweise für 30 und 60 km/h und durch bewegliche leuchtende Anzeigen für andere Geschwindigkeiten angezeigt. Die Geschwindigkeitsbegrenzungen für feste Zonen werden durch ortsfeste leuchtende Anzeigen oder durch ortsfeste reflektierende Schilder angezeigt.

*Fermé : Indication d'annonce* Geschlossen: Vorsignal



*Ralentissement 30 : R*

*commande de ne pas dépasser 30km/h au franchissement des aiguilles en aval du rappel de ralentissement 30 suivant*

Langsamfahrt 30: R

Befehl, die nächste direkt nach dem Erinnerungssignal liegende Weiche mit höchstens 30 km/h zu überfahren

*Ralentissement 60 : (R)*

*commande de ne pas dépasser 60km/h au franchissement des aiguilles en aval du rappel de ralentissement 60 suivant*

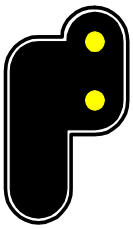
Langsamfahrt 60: (R)

Befehl, die nächste direkt nach dem Erinnerungssignal liegende Weiche mit höchstens 60 km/h zu überfahren

*Fermé : Indication fixe d'annonce pancarte en noir sur fond blanc.*

Geschlossen: Feste Geschwindigkeitsankündigungstafel in schwarz auf weißen Grund

*Fermé: Indication d'exécution* Geschlossen: Hauptsignal



*Rappel de Ralentissement 30 : RR*

*commande de ne pas dépasser 30km/h au franchissement des aiguilles en aval*

Erinnerung der Langsamfahrt 30: RR

Befehl, die nachgelagerte Weiche nur mit 30 km/h zu überfahren

*Rappel de Ralentissement 60 : (RR)*

*commande de ne pas dépasser 60km/h au franchissement des aiguilles en aval*

Erinnerung der Langsamfahrt 60: (RR)

Befehl, die nachgelagerte Weiche nur mit 60 km/h zu überfahren

*Indications d'exécution pancarte en blanc sur fond noir / Indication de libération.*

Hauptsignal: Feste Geschwindigkeitsankündigungstafel in weiß auf schwarzen Grund.

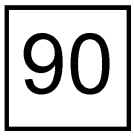



*Tableau Indicateur de Vitesse (TIV) ordinaire fixe:*

*Annonce d'une zone de limitation de vitesse*

Normale feste Geschwindigkeitsanzeigetafel (TIV); die Geschwindigkeit muss am Anfang des mit „Z" markierten Bereichs ausgeführt sein.


*Pancarte Z fixe*

*origine d'une zone fixe de limitation de vitesse*

Feste Tafel Z

Anfang eines Bereichs mit Geschwindigkeitsbegrenzung


*Pancarte R fixe*

*fin de la zone fixe de limitation de vitesse (Reprise)*

Feste Tafel R

Ende des Bereichs mit Geschwindigkeitsbegrenzung

*Indication mobile d'annonce pancarte en noir sur fond blanc.*

Bewegliche Anzeigetafel in schwarz auf weißem Grund.

*Bande blanche : Non présentation d'un Tableau indicateur de vitesse mobile d'annonce*

Weißes Band: Die Ankündigung der auf der beweglichen Anzeige angezeigten Geschwindigkeitsbegrenzung gilt nicht


*Tableau indicateur de vitesse sur pointe mobile : Annonce d'une zone de limitation de vitesse relative au franchissement en voie déviée d'un appareil de voie*

Bewegliche Anzeige für die Ankündigung einer Geschwindigkeitsbegrenzung für die Zungenspitzen: Ankündigung einer Geschwindigkeit, die beim Abbiegen auf der nächsten Weiche einzuhalten ist.

*Fermé et Mobile: Indication d'exécution pancarte en blanc sur fond noir.*

Geschlossen und beweglich: Ausführungstafel in weiß auf schwarzem Grund.

*Bande blanche : Non présentation d'un tableau indicateur de vitesse mobile d'exécution*

Weißes Band: Die Geschwindigkeitsbegrenzung, die angezeigt wird, muss nicht ausgeführt werden


*Tableau indicateur de vitesse d'exécution et mobile : Origine d'une zone de limitation de vitesse sur un appareil de voie*

Bewegliche Geschwindigkeitstafel: Anfang des Bereichs mit Geschwindigkeitsbegrenzung im Bereich einer Weiche

# ANNEXE B

# Petite histoire des postes d'aiguillage en France

# ANHANG B

# Kleine Geschichte der französischen Stellwerke

## B.1 Apparitions des enclenchements Vignier et Saxby

Les premiers signaux étaient manœuvrés à pied d'œuvre comme toutes les aiguilles, sans relation d'ordre entre les différents appareils d'une même zone, et donc les risques d'erreur n'étaient pas négligeables. Pierre Auguste VIGNIER (ouvrier menuisier à la Compagnie de l'Ouest) eut l'idée de rendre solidaires les leviers d'aiguille et de signaux des bifurcations.

**B1**: Poste – Stellwerk „VIGNIER 1897"

C'est en Grande-Bretagne que les développements de l'invention de VIGNIER seront les plus importantes : TYER, SAXBY HOOK (1860) et SAXBY FARMER (1869)… Le système SAXBY FARMER perdurera en France ; étant préféré pour les tables d'enclenchements des postes importants (jusqu'à 200 leviers) aux postes VIGNIER (16 à 25 leviers) étant utilisés dans les autres cas. M. BOURE, inspecteur principal imagina en 1897 un système de serrures rendant matériellement obligatoire l'application des consignes de cadenassement entre leviers de signaux d'une part, leviers d'aiguilles et taquets d'autre part. Ce type d'enclenchements, très répandu porte aujourd'hui le nom de " serrure d'enclenchement à clés S " ; il a permis de traiter les enclenchements des petites gares de façon économique. Après un siècle d'existence, les «serrures BOURE» restent complémentaires des systèmes modernes d'enclenchements.

## B.1 Entstehen der Stellwerkslogik Vignier und Saxby

Die ersten Signale waren ortsbedient, wie alle Weichen, ohne Reihenfolge zwischen den verschiedenen Geräten ein und derselben Zone. Die Fehlerrisiken waren so nicht unbedeutend. Pierre Auguste Vignier (Schreiner in der Gesellschaft des Westens - 1855) hatte die Idee, die Hebel der Weichen und der Signale der Abzweigungen zu verbinden. In Großbritannien fanden die wichtigsten Weiterentwicklungen der Erfindung von Vignier statt: Tyer, Saxby Hook (1860) und Saxby Farmer (1869). Das System Saxby Farmer wird in Frankreich fortbestehen; es ist für die Verschlussregister wichtiger Stellwerke (200 Hebel) den VignierStellwerken (16 bis 25 Hebel) vorzuziehen, die in den anderen Fällen benützt werden.

**B2**: Poste – Stellwerk „SAXBY 1900"

Hauptinspektor Bouré hat 1897 die Idee, ein System von Schlössern zu entwickeln, das physikalisch dazu zwingt die Verschlussvorschriften zwischen den Signalhebeln einerseits und den Weichenhebeln und Verschlussstücken andererseits einzuhalten. Dieser weit verbreitete Typ von Sicherung trägt heute den Namen „Schlüsselwerk mit S Schlüsseln"; er hat es erlaubt, die Sicherung kleiner Bahnhöfe auf eine wirtschaftliche Art und Weise umzusetzen. Nach einem Jahrhundert Existenz bleiben diese Schlüssel eine Ergänzung moderner Sicherungssysteme.

## B.2 Évolution des postes mécaniques

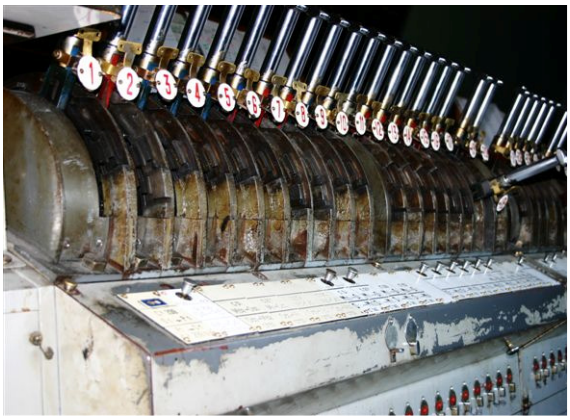
La SNCF mit au point avec les établissements SAXBY deux types de postes mécaniques intégrant la commande de signaux et d'aiguilles par des petits leviers. Ce sont :

- Le poste électromécanique (EMU), réalisé à 144 exemplaires entre 1946 et 1976 sa capacité maximale est de 180 leviers ;
- Le poste mécanique 1945 (MU 1945) ; 350 furent construits en 30 ans.

En 2008, on dénombre encore 570 postes mécaniques dont plus de 300 de type unifié, 200 de type SAXBY et 50 des types ex-AL, VIGNIER et TYER ; 350 gares, essentiellement de voie unique, essentiellement de voie unique, étant pourvues de serrures centrales d'enclenchement à clés S.

Les postes électriques à leviers d'itinéraires

L'utilisation de fluides levant toute restriction quant à l'effort de manœuvre, il était naturel de rechercher la commande globale des appareils d'un itinéraire. Le but est que chaque levier commande les aiguilles et les signaux d'un itinéraire. Les incompatibilités sont réalisées mécaniquement au niveau des organes de commande des appareils de voie.



**B3:** Poste – Stellwerk „Aster" (Strasbourg 1947)

Le premier poste à leviers d'itinéraires (COSSMANN-DUCOUSSO-BLEYNIE) fut mis en service à Bordeaux Saint-Jean P1 en 1903. ALBERT DESCUBES base le poste sur la topographie du plan des voies. Il définit un itinéraire à l'aide de la manœuvre de deux leviers correspondant, l'un à la destination de l'itinéraire et l'autre à son origine. Il conçut ensuite un système où les incompatibilités entre itinéraires sont assurées par des relais électromécaniques. Au total, 200 postes à leviers d'itinéraires auront été réalisés depuis 1903 ; il en subsistait 44 en 1997. Il est remarquable de noter que tous ces postes ont fait l'objet de conception et de validation formelle ! Les incompatibilités réalisées par ces postes sont écrites par un formalisme formel, non ambigu, permettant de vérifier l'existence et l'efficacité des enclenchements, réalisant les incompatibilités et traduisant ainsi les propriétés de sécurité à respecter.

## B.2 Entwicklung mechanischer Stellwerke

Die SNCF entwickelte zusammen mit den Saxby Niederlassungen zwei Typen mechanischer Stellwerke, die die Steuerung von Signalen (Lichtsignale) und von Weichen durch kleine Hebel ermöglichten. Es handelt sich um:

- Das elektromechanische Stellwerk, das zwischen 1946 und 1976, 144 Mal verwirklicht wurde (Höchstkapazität von 180 Hebeln).
- Das mechanische Stellwerk von 1945, das in 30 Jahren 350 Mal gebaut wurde.

Im Jahr 2008 gibt es immer noch 570 mechanische Stellwerke, darunter mehr als 300 vom vereinheitlichten Typ, 200 von der Marke Saxby und 50 vom Typ Ex-Al, Vignier und Tyer; 350 Bahnhöfe, hauptsächlich auf eingleisigen Strecken, sind mit zentralen S-Sicherungsschlössern ausgestattet.

Die elektrischen Fahrstraßenstellwerke

Die Benutzung von Flüssigkeiten, die jede Beschränkung der Umstellkraft aufheben, ermöglichte es, eine globale Steuerung aller Weichen einer Fahrstraße zu entwickeln. Ziel war es, dass ein einziger Hebel die Weichen und Signale einer Fahrstraße steuert. Die Unvereinbarkeit wird mechanisch im Bereich der Antriebsorgane der Weichen erzeugt. Das erste Stellwerk mit Fahrstraßenhebeln (Cossmann – Ducousso - Bleynie) wurde in Bordeaux Saint-Jean P1 im Jahre 1903 in Betrieb genommen. Albert Descubes, baut das Stellwerk auf dem Gleisplan beruhend auf. Er definiert die Fahrstraße mithilfe von zwei Hebeln, für das Ziel und den Ursprung der Strecke. Er entwickelte danach ein System, bei dem die Unvereinbarkeit zwischen Fahrstraßen durch elektromechanische Relais gewährleistet wird. Insgesamt wurden seit 1903 200 Stellwerke mit Fahrstraßenhebeln verwirklicht. 1997 gab es noch 44. Es ist bemerkenswert, dass all diese Stellwerke formal konzipiert und validiert wurden! Die durch diese Stellwerke umgesetzten Unvereinbarkeiten werden durch eine eindeutige formale Formulierung beschrieben, die es erlaubt, die Existenz und die Wirksamkeit der Verschlüsse zu prüfen, die die Unvereinbarkeiten verwirklichen und so die zu respektierenden Sicherheitseigenschaften übersetzt.

## B.3 Les postes tout relais de la SNCF

Dès la création de la SNCF sous l'impulsion de Jean WALTER, l'étude de postes sans enclenchements mécaniques ou électromécaniques fut entreprise. L'électrification de Paris Lyon sera un véritable champ d'essai des postes tout relais avec la mise en service (1949) des Postes tout Relais à destruction Automatique des itinéraires et à transits souples. Il constitue le père de la grande famille des Postes tout Relais à transit Souple (PRS) dont le premier fut celui de Gagny (1950). Leur développement a été important dans la période de 1960 à 1980.

Conséquence d'un concours lancé en 1968, apparaît le Poste tout Relais Géographique (PRG), dont la caractéristique visible est la commande d'itinéraire par action sur deux boutons d'une table à tracé géographique correspondant aux extrémités de l'itinéraire et le principe technologique de constituer des ensembles correspondant aux éléments du plan des voies (aiguilles, signaux, circuits de voie…).
De 1949 à nos jours, plus de 600 PRS ont été installés ; le nombre de ceux en service en 2008 est d'environ 500, valeur stable depuis une décennie, et les PRG sont au nombre de 113.

## B.4 L'informatique dans les postes d'aiguillage

L'informatique a été introduite dans la signalisation entre 1970 et 1980 par la mise en œuvre de programmateurs, offrant une aide à l'exploitation des postes. PRS et PRG ayant chacun leurs partisans, le rôle d'arbitre fut dévolu au Poste à Relais et à Commande Informatisé (PRCI) apparu en 1983 à la Ferté-Alais ; celui-ci associe un étage informatique de programmation des itinéraires assurant également la commande et la formation des itinéraires. Les enclenchements et les commandes des appareils sur le terrain restent le domaine d'une structure géographique, identique à celle du PRG.

Le PRCI dont 220 exemplaires sont en service, a pris le relève de PRS dans de nombreuses grandes gares (Paris Montparnasse, Paris Nord, Paris Est, Thionville, Lille…) en attendant d'être lui-même victime du « tout informatique ».

Les premiers systèmes d'enclenchements informatiques Modulaires d'Équipement des Lignes à voie unique (SYMEL) [Antoni, 2005] [Antoni, 2009-1] sont mis en service en juin 1995. C'est la base du futur PIPC.

## B.3 Relaisstellwerke der SNCF

Nach der Gründung der SNCF wurde unter der Initiative von Jean Walter eine Studie über Stellwerke ohne mechanische oder elektromechanische Verschlüsse durchgeführt. Die Elektrifizierung der Strecke Paris-Lyon wurde ein echtes Versuchsfeld für Relaisstellwerke mit der Inbetriebnahme des Relaisstellwerks mit automatischer Fahrstraßenauflösung und Teilauflösung 1949. Dies wurde der Prototyp der großen Familie der Stellwerke mit Fahrstraßenteilauflösung (PRS), wobei das erste davon 1950 in Gagny in Betrieb genommen wurde. Ihre Entwicklung fand vor allem in der Periode von 1960 bis 1980 statt.
Nach einem Wettbewerb im Jahre 1968, entsteht das geographische Relaisstellwerk (PRG), dessen sichtbare Eigenschaft die Fahrstraßensteuerung durch Drücken von zwei Knöpfen ist. Diese Knöpfe befinden sich auf einem geographischen Gleisbildstellpult und entsprechen den beiden Enden der Fahrstraße. Eine andere Eigenschaft ist der Grundsatz, alle Elemente des Gleisplans darzustellen (Weichen, Signale, Gleisstromkreise). Seit 1949 wurden 600 PRS installiert und 2008 waren ungefähr noch 500 in Betrieb, seit Jahrzehnten ein konstanter Wert. Es gibt 113 PRG-Stellwerke.

## B.4 Rechner in den Stellwerken

Die Informatik hielt zwischen 1970 und 1980 in den Stellwerken Einzug und zwar durch die Inbetriebnahme von automatischen Programmen, die beim Betrieb des Stellwerks helfen. Da es sowohl Befürworter des PRS als auch des PRG gab, wurde als Kompromiss das Relaisstellwerk mit rechnerunterstützter Steuerung (PRCI) entwickelt und 1983 in Ferté-Alais eingebaut. Dieses Stellwerk besitzt eine Rechnerebene für die Programmierung der Fahrstraßen und ebenso für deren Steuerung und Bildung. Die Sicherung und die Steuerung der Weichen vor Ort werden immer noch mithilfe einer geographischen Relaisstruktur durchgeführt, die der des PRG-Stellwerks entspricht.
Das PRCI, von dem 220 Exemplare im Dienst sind, ist in zahlreichen großen Bahnhöfen (Paris-Montparnasse, Paris-Nord, Paris-Est, Thionville, Lille…) der Nachfolger des PRS. Diese Stellwerke warten darauf, durch komplett rechnergestützte Stellwerke ersetzt zu werden.
Das erste modulare Verschlussrechnersystem für die Ausstattung von eingleisigen Stecken (SYMEL) [Antoni, 2005] [Antoni, 2009-1] wurde in Juni 1995 in Betrieb genommen. Es ist die Grundlage für den später folgenden PIPC.

Enfin, le premier Poste d'Aiguillage Informatique (PAI), a été mis en service en juillet 1997 à Roanne. Il devrait, ouvrir en France, le règne des postes « tout informatique » qui existaient déjà à l'étranger (suède, Grande-Bretagne) depuis plus de dix ans. L'informatique réalise l'interface avec l'exploitant ; elle permet l'enrichissement des aides apportées, l'extension de la zone d'action des opérateurs et par là même génère des gains en termes de productivité et de souplesse d'exploitation. L'objectif poursuivi étant une diminution des coûts en conservant un niveau de sécurité analogue à celui des postes à relais.

## B.5 Les commandes centralisées

La commande à distance de postes d'aiguillage tarda à se développer en France. La première réalisation eut lieu en 1933, gérant une voie banalisée à partir du poste de commandement de Paris Saint Lazare. L'ère des télétransmissions électroniques sera inaugurée en 1958 sur la commande de la voie unique Dôle Vallorbe [SNCF, 1963]. Enfin l'électronique cédera le pas à l'informatique à partir de 1980 dans ces types d'équipements, et en 1985 sera réalisé un Système Normalisé de Télétransmissions Informatiques (SNTI), qui lui-même est désormais concurrencé par l'établissement de liaisons directes entre microprocesseurs de commandes des postes.

## B.6 L'évolution technologique

On pourrait ainsi déterminer les ères technologiques de la signalisation [Gernigon, 1998] : D'abord mécanique et électromagnétique puis électrodynamique au XIX$^{ème}$ siècle, électrique, et automatique, puis électronique et enfin informatique aujourd'hui. Remarquons que ceci n'interdit pas d'envisager le cas où les postes d'aiguillages s'interfacent avec d'autres systèmes conçus eux-mêmes comme un tout (avec un contrôle de vitesse par exemple, auquel cas l'action du poste d'aiguillage se prolonge jusqu'à l'action directe sur la marche du train). Le nombre de postes mécaniques diminue régulièrement par suite du développement des postes électriques, PRCI notamment, et des fermetures de ligne.

De façon générale, les zones à fort trafic sont gérées par les postes électriques de type PRS, PRG ou PRCI. Progressivement le PRS est remplacé par le PRCI ou le PAI.

Schließlich ist das erste Informatikstellwerk (PAI), im Juli 1997 in Roanne in Dienst gegangen. Es sollte in Frankreich das Zeitalter des „komplett mit Informatik" eröffnen, das schon im Ausland (Schweden, Großbritannien) seit mehr als zehn Jahren besteht. Die Informatik verwirklicht die Schnittstelle mit dem Betriebspersonal; sie erlaubt mehr Hilfsfunktionen, die Ausweitung des Eingriffsbereichs des Betriebspersonals. Auf gleiche Weise gewinnt man bei der Produktivität und der Betriebsflexibilität. Die verfolgte Zielsetzung ist die einer Kostensenkung bei gleichzeitiger Beibehaltung eines Sicherheitsniveaus, das dem der Relaisstellwerke entspricht.

## B.5 Zentrale Steuerung

Die Fernsteuerung von Stellwerken hat sich in Frankreich sehr spät entwickelt. Die erste Umsetzung fand im Jahre 1933 statt, für den Betrieb einer Strecke mit Fahrt in Gegenrichtung von der Betriebzentrale in Paris-Saint-Lazare aus. Das Zeitalter der elektronischen Fernübertragungen wird im Jahre 1958 mit der Steuerung der eingleisigen Strecke Dôle-Vallorbe [SNCF, 1963] eröffnet. Ab 1980 löst die Informatik die Elektronik bei solchen Arten von Stellwerken ab und im Jahre 1985 wird ein genormtes System mit Informatikfernübertragung entwickelt. Diesem System wird wiederum, mit der Schaffung direkter Verbindungen zwischen den Mikroprozessoren der Steuerung und den Stellwerken, Konkurrenz gemacht.

## B.6 Technologische Entwicklung

Die technologischen Zeitalter der Signalgebung lassen sich folgendermaßen definieren [Gernigon, 1998]: Zuerst mechanisch und elektromagnetisch dann, 19. Jahrhundert elektrodynamisch, elektrisch und automatisch dann elektronisch und schließlich heutzutage rechnergestützt. Es ist anzumerken, dass dies nicht verbietet, den Fall in Betracht zu ziehen, bei dem die Stellwerke mit anderen Systemen Schnittstellen haben die selbst als ein System darstellen (mit einer Geschwindigkeitskontrolle zum Beispiel, in welchen Fall der Handlungsradius des Stellwerks bis auf die direkte Beeinflussung der Züge ausgeweitet wird). Die Anzahl der mechanischen Stellwerke geht infolge der Entwicklung elektrischen Stellwerke insbesondere des PRCI und von Streckenschließungen regelmäßig zurück. Allgemein werden die Zonen mit starkem Verkehr durch elektrische Stellwerke des Typs PRS, PRG oder PRCI verwaltet. Progressiv wird das PRS durch das PRCI oder das PAI ersetzt.

## B.6.1 PRS : Poste tout Relais à transit Souple

1ère mise en service en 1952, Lyon-Perrache : l'aiguilleur dispose d'une table de commande à boutons d'itinéraires et d'un TCO.

Le fonctionnement du poste est entièrement assuré au moyen de circuits électriques (à l'exclusion d'enclenchements mécaniques).

**B4:** Poste – Stellwerk „PRS" (Sélestat) 1969

## B.6.1 PRS: Relaisstellwerk mit Teilauflösung

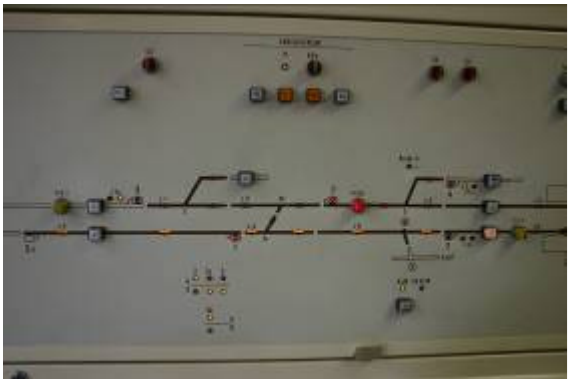
Erste Inbetriebnahme im Jahre 1952 in Lyon-Perrache: Der Weichensteller verfügt über eine Stelltafel mit Fahrstraßenknöpfen und einer Überwachungsanzeige (TCO).

Das Funktionieren des Stellwerkes wird völlig durch elektrische Schaltkreise sichergestellt (Ausnahme: die mechanischen Verschlüsse).

## B.6.2 PRG : Poste tout Relais à câblage Géographique

1ère mise en service en 1971, Gretz et Montargis : les organes de commande et les voyants de contrôle sont installés sur une table de commande et de contrôle (TCC) unique. La commande d'un itinéraire s'effectue en pressant successivement sur le bouton de destination, puis sur celui d'origine. Le fonctionnement du poste est entièrement assuré au moyen de circuits électriques (à l'exclusion d'enclenchements mécaniques).

Le PRG est dit géographique parce que les circuits électriques sont basés sur une structure géographique reproduisant le tracé des voies et aussi parce que les organes de commande (boutons, commutateurs…) sont habituellement installés sur un tracé géographique des voies, représenté sur le plan de travail (TCC).



**B5:** Poste – Stellwerk „PRG" (Amiens 1980)

## B.6.2 PRG : Stellwerke mit geographischen Kabelverbindungsrelais

Erste Inbetriebnahme im Jahre 1971 in Gretz und Montargis: Die Steuerungsorgane und die Kontrolllichter werden auf einer einzigen Steuer- und Kontrolltafel (TCC) installiert. Die Steuerung einer Fahrstraße wird durchgeführt, indem man erst auf das Ziel und dann auf den Ursprung drückt. Das Funktionieren des Stellwerks ist völlig mithilfe von Stromkreise sichergestellt (Ausnahme: die mechanischen Verschlüsse).

Das PRG wird geographisch genannt, weil die Stromkreise auf einer geographischen Struktur basieren, die die Gleise reproduziert. Zudem sind die Steuerorgane (Knöpfe, Schalter…) gewöhnlich auf einer geographischen Darstellung der Gleise installiert, die auf der Gleisbildstelltafel dargestellt werden (TCC).

## B.6.3 PRCI : Poste à Relais et à Commande Informatique

1ère mise en service en 1983, La Ferté-Alais : l'aiguilleur commande les itinéraires à l'aide d'un clavier informatique. Les contrôles s'effectuent soit sur le TCO soit sur une console de visualisation (image écran).

***B6:*** *Poste – Stellwerk „PRCI" (Paris Montparnasse)*

Le fonctionnement du poste est assuré au moyen :

- d'un système informatique qui réalise toutes les fonctions ne présentant pas un haut niveau de sécurité (préparation…) ;
- de circuits électriques (avec relais de sécurité) dont le rôle est essentiellement de réaliser les enclenchements et l'ouverture des signaux

## B.6.4 PAI : Postes d'Aiguillage Informatiques

1ère mise en service en 1997, Roanne : le PAI est un poste dans lequel les fonctions d'enclenchement ne sont pas réalisées par une logique câblée à base de relais de sécurité, mais par une logique programmée à base de microprocesseurs. Le PAI comporte deux niveaux :

- Le premier niveau est constitué par un équipement de commande informatique d'itinéraires identique à celui du PRCI ;
- Le deuxième niveau est constitué par les modules d'enclenchement qui assurent véritablement les fonctions de sécurité.

Le PAI est télécommandable et compatible avec tous les modules d'aide à l'exploitation. Le PAI est équipé soit d'un TCO classique soit d'un TCO sur écran. Le temps alloué aux essais avant mise en service et, le cas échéant, aux études de modification des fonctions du poste, sont sensiblement plus longues qu'avec des postes à relais.

## B.6.3 PRCI: Rechnergesteuertes Relaisstellwerk

Erste Inbetriebnahme 1983 in La-Ferté-Alais: Der Weichensteller steuert Fahrstraßen mittels einer Computertastatur. Die Kontrollen werden entweder auf der Gleistafel oder auf einer Sichtanzeige (Bildschirm) durchgeführt.

Die Funktion des Stellwerks wird gewährleistet durch:

- ein Informatiksystem (Rechner), das alle Funktionen umsetzt, die kein hohes Sicherheitsniveau benötigen (Vorbereitung, Bedienung…).
- Stromkreise (mit Sicherungsrelais), deren Rolle es hauptsächlich ist, das Einrasten und die Öffnung der Signale durchzuführen.

## B.6.4 PAI: Rechnerbasiertes Stellwerk

Erste Inbetriebnahme 1997 in Roanne. Das PAI ist ein Stellwerk, in dem die Sicherungsfunktionen nicht durch eine verkabelte Logik aus Sicherungsrelais, sondern durch eine programmierte Mikroprozessorlogik verwirklicht werden. Das PAI umfasst zwei Ebenen:

- Die erste besteht aus einem Rechnersystem zur Fahrstraßeneinstellung, das mit jenem des PRCI identisch ist.
- Die zweite besteht aus Sicherungsmodulen die die eigentlichen Sicherheitsfunktionen gewährleisten.

Das PAI ist ferngesteuert und mit allen Hilfsmodulen für den Betrieb kompatibel. Das PAI ist entweder mit einer klassischen Gleisbildtafel (TCO) oder mit einer Bildschirmgleisbildtafel ausgestattet. Die für Tests vor Inbetriebnahme und eventuelle Studien zu Stellwerksfunktionsänderungen nötige Zeit ist erheblich länger.

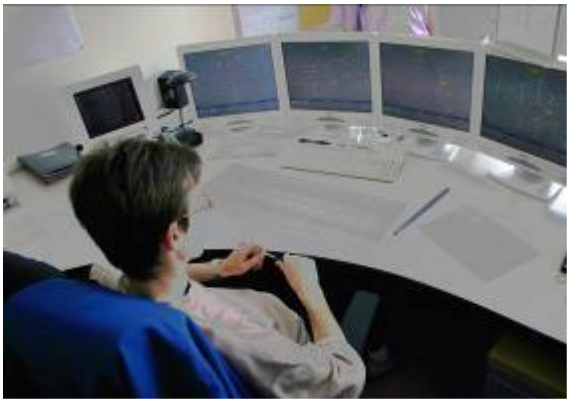
## B.6.5 PIPC : Poste Informatique avec PC

1ère mise en service en 1999 (Moirans–Saint Marcellin) pour les PIPC, mise en service projeté en 2010 pour les PAI de nouvelle génération. Faisant suite au système de gestion des lignes SYMEL (1995), le PIPC est un poste dans lequel les fonctions d'enclenchement ne sont pas réalisées par une logique câblée à base de relais de sécurité, mais par une logique programmée à base de PC industriels.

Le PIPC est télécommandable et compatible avec tous les modules d'aide à l'exploitation. Actuellement, 160 postes en service en 2008.

**Le PIPC est le seul poste d'aiguillage français**

**B7**: Poste - Stellwerk „PIPC" (Toulouse) 1998 [SNCF, 2002]



conçu pour être aujourd'hui prouvé formellement. C'est le poste d'aiguillage faisant l'objet de ce travail. Ses principes ont été retenus et imposés pour conception de nouveaux postes d'aiguillage français à compter de 2010 (PAING: Poste Informatique de Nouvelle Génération).

## B.6.5 PIPC: Rechnerstellwerk mit PC-Technologie

Die erste Inbetriebnahme für das PIPC 1997 fand in Moirans–Saint Marcellin statt, die Inbetriebnahme des PAI der neuen Generation ist für 2010 geplant. Das PIPC Stellwerk ist eine Weitentwicklung des Streckensteuerungssystems SYMEL (1995). Die Sicherungsfunktionen sind nicht durch eine verkabelte Logik aus Sicherheitsrelais, sondern durch eine auf Industrie-PCs programmierte Logik verwirklicht.

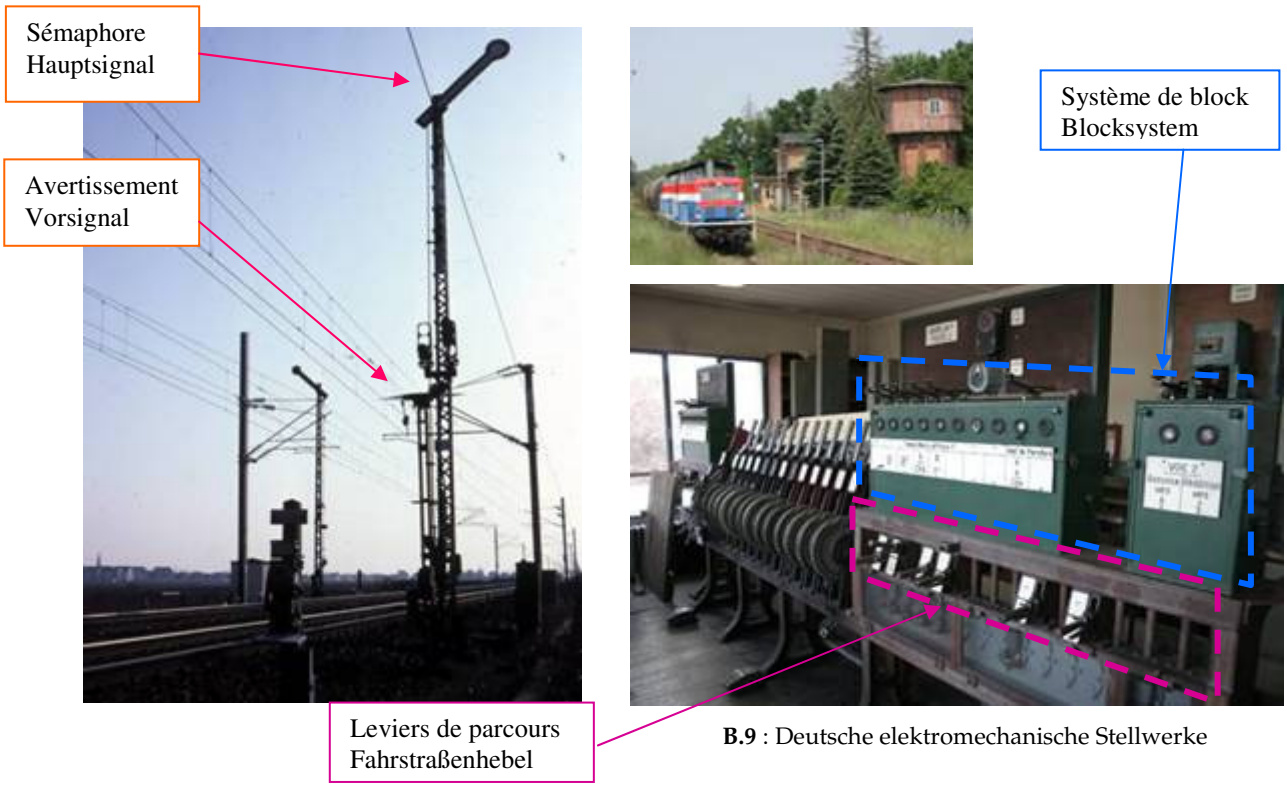
Das PIPC ist ferngesteuert und mit allen Hilfsmodulen für den Betrieb kompatibel. 2008 waren 160 solcher Stellwerke in Betrieb.

**B8:** Poste – Stellwerk „MEI" 1998 [SNCF, 2002]

Das PIPC ist das einzige französische Stellwerk, das geplant wurde um formell bewiesen zu werden. Es ist somit Gegenstand dieser Arbeit. Seine Grundsätze wurden für die Konzeption neuer Stellwerke in Frankreich ab 2010 ausgewählt und vorgeschrieben (PAING: Rechnerstellwerk neuer Generation).

# B.7 Comparaison des postes d'aiguillage français et allemands

## B.7.1 Postes électromécaniques allemands




**B.9** : Deutsche elektromechanische Stellwerke

Processus simplifié:

- Manœuvre libre des aiguilles (sans ordre de manœuvre) des leviers non enclenchés par un levier de parcours
  → Commande mécanique ou électrique des aiguilles + lanternes
- Manœuvre du levier de parcours A vers B :
  → enclenchement mécaniquement les leviers d'aiguille
- Manœuvre du levier de signal de protection (S) si les conditions de block sont correctes (An)
  → anti-répétiteur du levier du signal
  → mise en œuvre du transit rigide sur le levier de parcours [ITI formé]
  → Commande mécanique ou électrique du signal (si mécanique, lancement des verrous de pointe)

# B.7 Vergleich zwischen französischen und deutschen Stellwerken

## B.7.1 Deutsche elektromechanische Stellwerke

Vereinfachter Prozess:

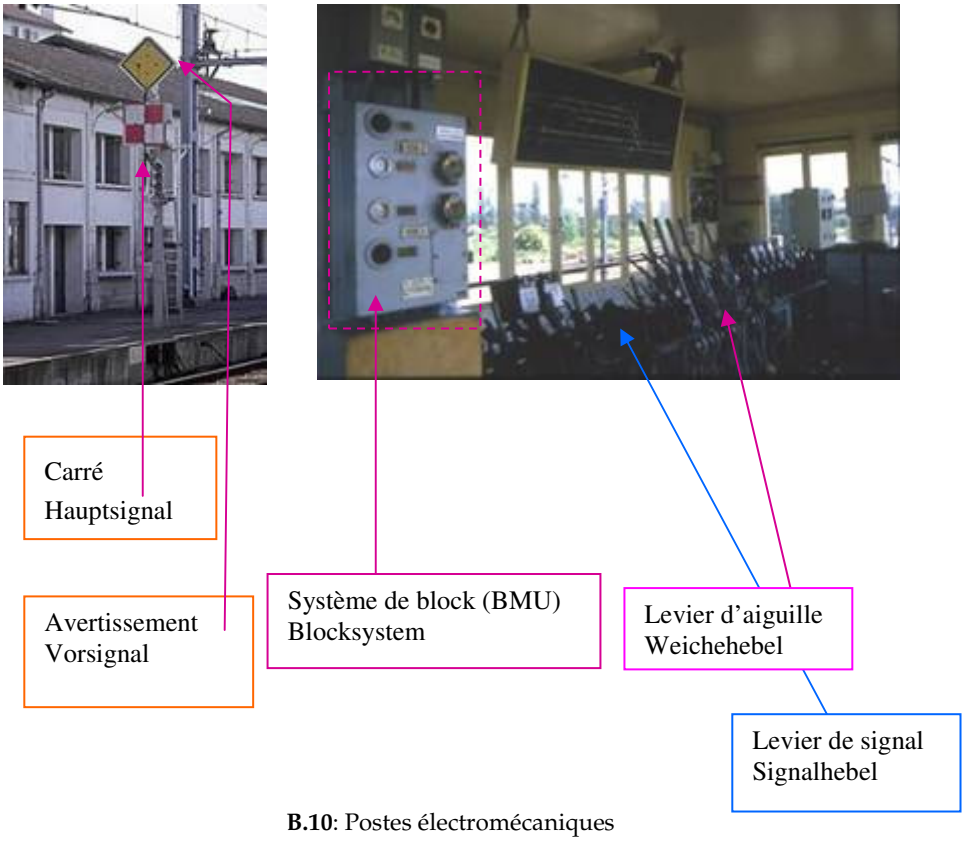
- Freie Bewegung der Weichenhebel (ohne Reihenfolge), nicht durch einen Fahrstraßehebel verschlossen
  → mechanische oder elektrische Steuerung der Weiche + Laterne
- Bewegung der Fahrstraßehebel von A nach B
  → mechanische Weichensicherung
- Stellen des Deckungssignalhebels (S) wenn der Block frei ist (An)
  → Wiederholungssperre („Hampelmann")
  → Einstellen der festen Fahrstraße auf der keine Teilauflösung möglich ist (Fahrstraße gebildet)
  → mechanische oder elektrische Steuerung des Hauptsignals (falls mechanisch, Aktivierung der Zungenriegel)

- Ouverture du signal de protection si toutes les conditions de contrôles sont OK ;
- Le train franchit le signal
  → la fermeture du signal (protection ET block absolu) peut être commandée ;
  → le signal ne peut être ré ouvert que la reddition voie libre a été effectuée ET qu'une nouvelle annonce a été reçue ;
- Le train libère une à une les aiguilles (sans effet) ;
- Le train a dégagé la dernière aiguille de l'itinéraire (action sur un détecteur en extrémité de zone) :
  → si le signal est fermé, possibilité de remettre en position neutre le levier de parcours ;
  → libération de toutes les aiguilles de l'itinéraire.

## B.7.2 Postes électromécaniques français

- Öffnung des Hauptsignals falls alle Kontrollbedingungen erfüllt sind
- Der Zug fährt über das Signal
  → Das Signal kann wieder geschlossen werden (Deckung und absoluter Block).
  → Das Signal kann erst wieder geöffnet werden wenn der Block frei gemacht wurde UND eine neue Meldung angekommen ist
- Der Zug macht nach und nach die Weichen frei (alles bleibt geschlossen).
- Der Zug befreit die letzte Weiche der Fahrstraße:
  → wenn der Signal schon geschlossen ist, Möglichkeit den Fahrstraßenhebel in die freie Position zurückzustellen
  → Alle Weiche sind wieder frei.

## B.7.2 Französische elektromechanische Stellwerke




**B.10**: Postes électromécaniques français - Französische elektromechanische Stellwerke

Processus simplifié :

- Manœuvre des aiguilles dans l'ordre imposé par le poste des leviers non enclenchés par les zones de CdV :
  → Commande mécanique ou électrique des aiguilles ;
- Manœuvre du levier de signal de protection → levier du signal soumis à la Zone d'Approche ⇒ mise œuvre du transit souple ;
  → Commande mécanique ou électrique du signal ;
- Ouverture du signal de protection si toutes les conditions de contrôles sont OK ;
- Manœuvre du levier de signal de Block (S) si le canton suivant est libre ;
- Le train franchit le signal :
  → Fermeture automatique du signal de protection (C) (nécessaire car block permissif) ;
  → Confirmation de la fermeture du signal de protection ;
- Le train libère une à une les aiguilles :
  → libération progressive des aiguilles dégagées de l'itinéraire.

## B.7.3 Poste à itinéraire français

Caractéristiques spécifiques France

- Block permissif ;
- Distinction des signaux de protection et de block ;
- Fermeture automatique des signaux ;
- Transit souple (libération progressive des aiguilles) ;
- Défaillances non sûres des circuits de voie sont dangereuses ;
- Le mécanicien peut manœuvre les aiguilles à pied d'œuvre en secours…

Caractéristiques spécifiques Allemagne

- Block absolu ;
- Pas de distinction les signaux de protection et de block (S) ;
- Fermeture non automatique des signaux ;
- Transit rigide (libération en bloc des aiguilles) ;
- Défaillances non sûres des compteurs d'essieux (chute hors zone d'influence) sont dangereuses ;
- Le mécanicien ne peut manœuvre les aiguilles à pied d'œuvre même en secours…

Vereinfachter Prozess:

- Bewegung der Weichenhebel falls die Gleisstromkreise frei sind in der vom Stellwerk vorgegebenen Reihenfolge:
  → mechanische oder elektrische Steuerung der Weiche
- Betätigung des Deckungssignalhebels (C)
  → der Signalhebel untersteht dem Annäherungsabschnitt ⇒ Einstellung der Fahrstraße mit Teilauflösung
  → mechanische oder elektrische Steuerung des Signals
- Öffnung des Deckungssignals wenn alle Kontrollbedingungen erfüllt sind
- Umstellen des Blocksignalhebels (S) falls der Block frei ist
- Der Zug fährt über das Signal:
  → Das Deckungssignal (C) schließt automatisch (dies ist nötig da Block mit bedingten Haltsignalen)
  → Bestätigung der Schließung des Deckungssignals
- Der Zug befreit jede Weiche einzeln:
  → nach und nach werden die sich in der teilaufgelösten Fahrstraße befindlichen Weichen freigemacht. Die Weichen sind frei wenn ihre Gleisstromkreise überfahren und freigemacht sind.

## B.7.3 Französische Fahrstrassen Stellwerke

Spezifische Eigenschaften in Frankreich

- Block mit bedingten Haltsignalen
- Deckungssignal und Blocksignal sind unterschiedlich
- Signale gehen automatisch zu
- Fahrstraße mit Teilauflösung
- unsicherer Ausfall des Gleisstromkreises kann gefährlich sein
- Der Lokführer kann eine gestörte Weiche per Hand umlegen.

Spezifische Eigenschaften in Deutschland

- Unbedingte Haltsignale
- Deckungssignal ist auch Blocksignal
- Normaleweise gehen die Signale nicht automatisch zu.
- Fahrstraßen ohne Teilauflösung (alle Weiche werden auf einmal freigegeben)
- unsichere Ausfälle der Achszähler sind gefährlich (Abschalten des Achszählers außerhalb der betreffenden Zone)
- Der Lokführer kann nie eine Weiche per Hand umlegen (selbst bei Störungen).

## B.8 Enseignements pour notre travail

C'est vers le milieu du XIX$^{ème}$ siècle, que Pierre-Auguste VIGNIER, a l'idée de mettre à profit le rapprochement des leviers pour réaliser les premiers enclenchements. Chaque levier actionne, en plus du signal ou de l'aiguille, le pêne d'un verrou qui fait obstacle à la manoeuvre d'un ou plusieurs leviers voisins s'il n'est pas dans la bonne position…

L'enclenchement est réciproque et interdit certaines combinaisons de la position des leviers, donc des aiguilles et signaux. Il est à noter qu'il ne s'agit pas d'un automatisme, au sens habituel du terme.

La mission dévolue à la technique n'est pas d'assister l'homme en réalisant de façon automatique des tâches pénibles ou répétitives. Elle lui complique le travail plus qu'elle ne l'allège. Sa fonction est de se prémunir de l'erreur humaine, mais sans pour autant remplacer l'homme. Elle automatise une fonction de contrôle auparavant dévolue à l'homme et laisse à l'homme la fonction d'exécution, c'est une fonction de contrainte.

## B.8 Erkenntnis für diese Arbeit

Mitte des 19. Jahrhundert hatte Pierre-Auguste Vignier die Idee, Hebel untereinander zu verbinden, um damit die ersten Verschlüsse herzustellen. Jeder Hebel bewegt, zusätzlich zum Signal oder der Weiche, den Riegel eines Schlosses, der einen oder mehrere Nachbarhebel blockiert, wenn er nicht in der richtigen Position ist.

Der Verschluss ist wechselseitig und verhindert bestimmte Stellkombinationen der Hebel, also der Weichen und der Signale. Es ist anzumerken, dass es sich hierbei nicht um eine Regelung in eigentlicher Form handelt.

Es ist nicht Aufgabe der Technik, dem Menschen durch automatische Ausführung unangenehmer oder sich wiederholender Aufgaben zu helfen. Die Technik erschwert die Arbeit mehr, als dass sie sie erleichtert. Ihre Aufgabe ist es, dem menschlichen Fehler vorzubeugen ohne jedoch den Menschen zu ersetzen. Sie automatisiert die Kontrollfunktionen, die früher vom Menschen durchgeführt wurden und überlässt ihm die Ausführung: die Technik hat also eine Beschränkungsfunktion.

Les premiers automatismes, au sens habituel, apparaîtront avec les postes d'aiguillage à *itinéraires* en 1929, généralisés en France que très progressivement après 1945.

A la fin des années 1980 aucun poste d'aiguillage en France ne fait appel à l'ordinateur pour piloter directement les signaux ou les moteurs d'aiguilles. Dans les postes informatisés, l'ordinateur est une couche supplémentaire entre l'aiguilleur et les aiguilles, mais il n'est pas le coeur du poste : il n'assure pas les *enclenchements*.

Le premier automatisme totalement informatisé, SYMEL, est mis en service seulement en 1995.

Les raisons de cette lenteur de mise en application en France sont simples : Dans le cas d'un relais électromécanique ou un levier, on sait ce qui se passe quand il tombe en panne. Un ordinateur, on ne savait pas très bien... L'informatisation des commandes pouvait simplifier grandement le travail des aiguilleurs… mais l'idée de confier un rôle de sécurité à l'ordinateur était rejetée car non compatible avec le paradigme des sécurités intrinsèques.

La sécurité intrinsèque est donc étroitement liée au principe déterministe. Mais elle laisse aussi, de fait, l'homme entièrement responsable de la sécurité. L'automatisme en défaut «passe la main» à l'homme.

Die ersten Regelungen im eigentlichen Sinn, tauchen mit den Fahrstraßenstellwerken 1929 auf, die in Frankreich nach 1945 allmählich verallgemeinert wurden.

Ende der 80er Jahre gibt es in Frankreich kein einziges Stellwerk, das Rechner benützt um Signale oder Weichenmotoren direkt anzusteuern. In den Rechnerstellwerken stellt der Rechner eine zusätzliche Schicht zwischen Weichensteller und Weichen dar, er ist aber nicht das Herzstück des Stellwerks: der Rechner führt keine Sicherungen durch.

Die erste vollständig rechnergestützte Steuerung SYMEL wird erst 1995 in Betrieb genommen.

Die Gründe für diese verspätete Einführung von Rechnern in Frankreich sind einfach: bei einem elektromechanischen Relais oder einem Hebel kennt man die Auswirkungen des Ausfall, bei einem Rechner kannte man sie nicht.

Die Einführung einer rechnergestützten Steuerung kann die Arbeit des Weichenstellers stark vereinfachen, aber die Idee, die Sicherheit einem Rechner anzuvertrauen wurde abgelehnt, da sie nicht dem Paradigma der systemintegrierten Sicherheit entsprach.

Die systemintegrierte Sicherheit ist also stark mit dem deterministischen Prinzip verbunden. Aber sie lässt den Menschen allein für die Sicherheit verantwortlich. Eine gestörte Regelung übergibt ihre Aufgabe an den Menschen.

# ANNEXE C

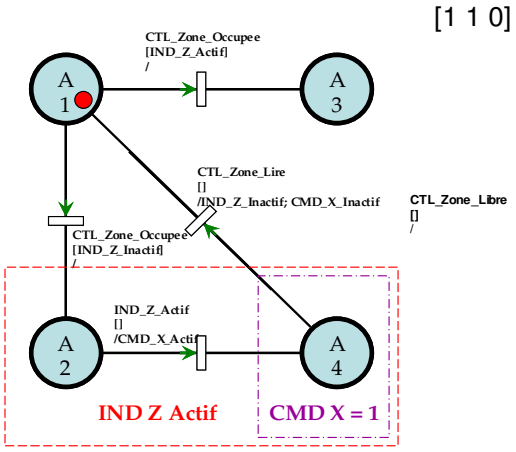
# Applications du langage AEFD à des cas simples

# ANHANG C

# Anwendung der AEFD-Sprache auf einfache Fälle

# C.1 Exemple 1 écrit en langage AEFD

## C.1.1 – État initial du système

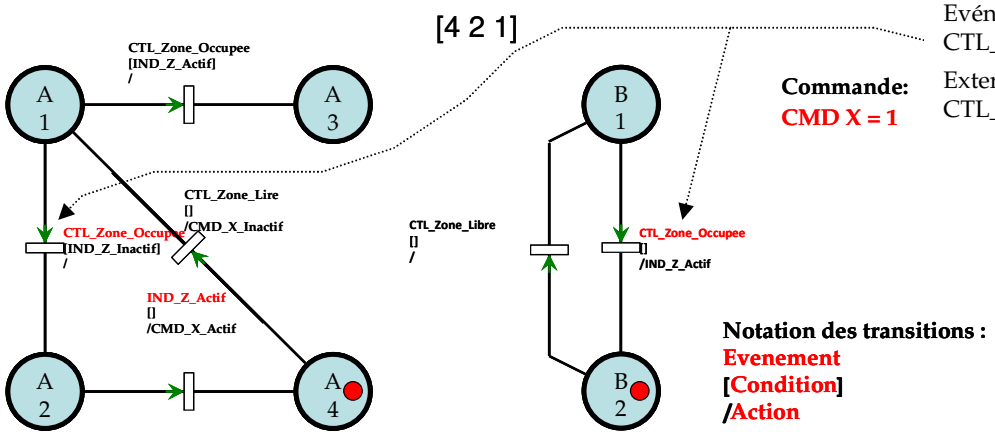



# C.1 Beispiel eins, in AEFD-Sprache geschrieben

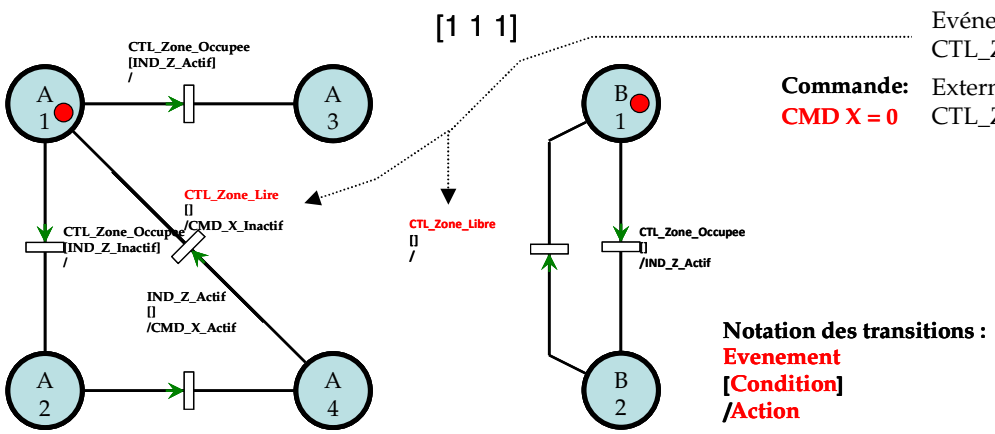
## C.1.1 – Anfangszustand des Systems

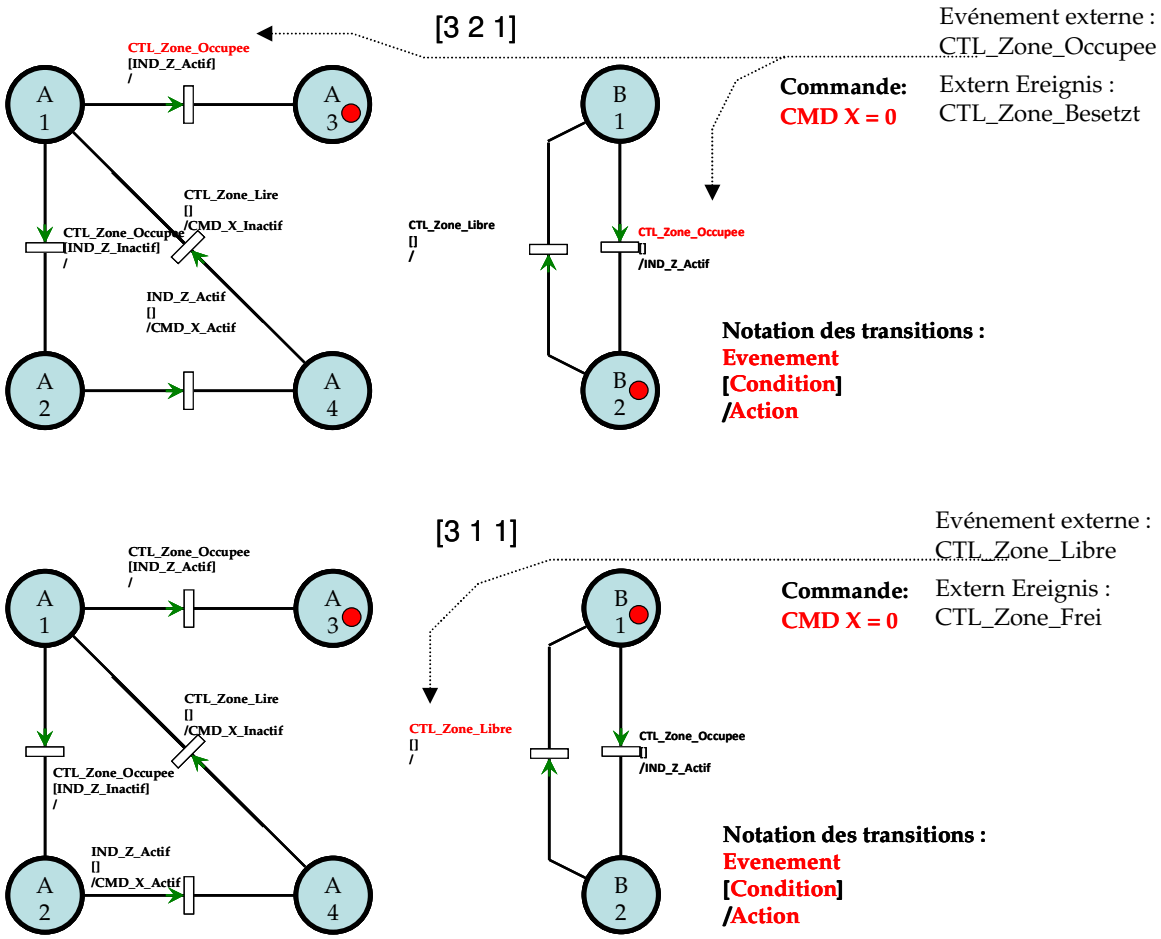
Es gilt überall:
- Événement = Ereignis
- Condition = Bedingung
- Action = Handlung
- Actif = Aktiv
- Occupé = Belegt
- Notation des transitions = Beschreibung der Transitionen
- Commande = Befehl
- Libre = Frei

## C.1.2 – États fonctionnels du système



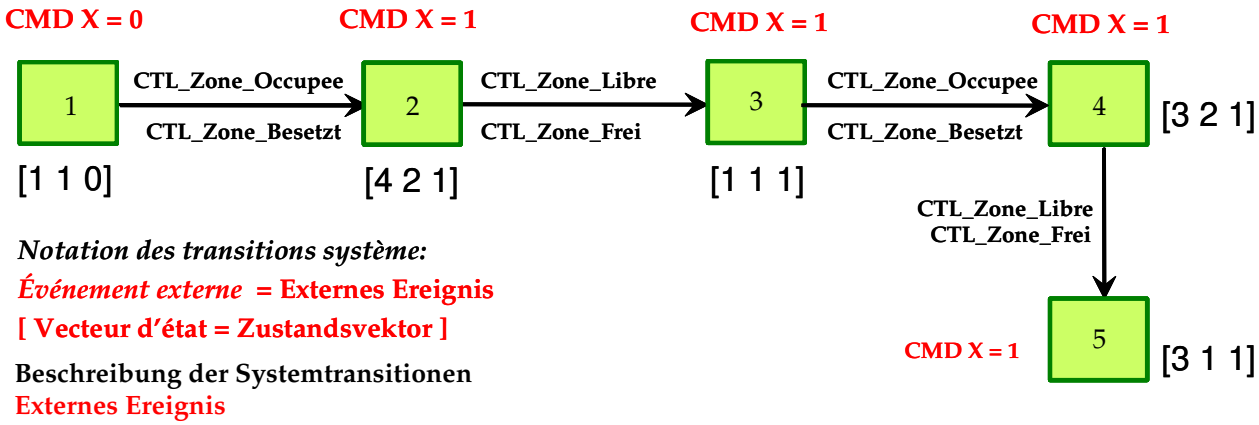

## C.1.2 – Funktioneller Systemzustand






## C.1.3 – Arbre des états système

## C.1.3 – Zustandsbaum des Systems

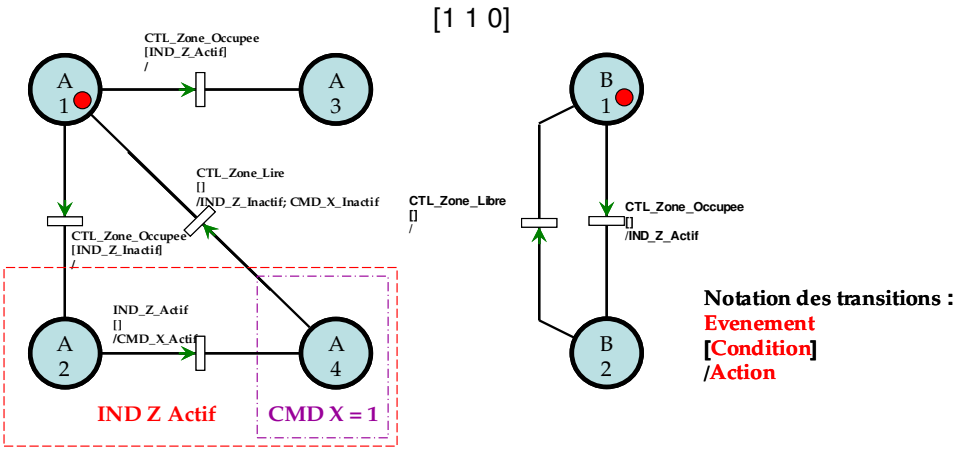



*Notation des transitions système:*
*Événement externe* = Externes Ereignis
[ Vecteur d'état = Zustandsvektor ]

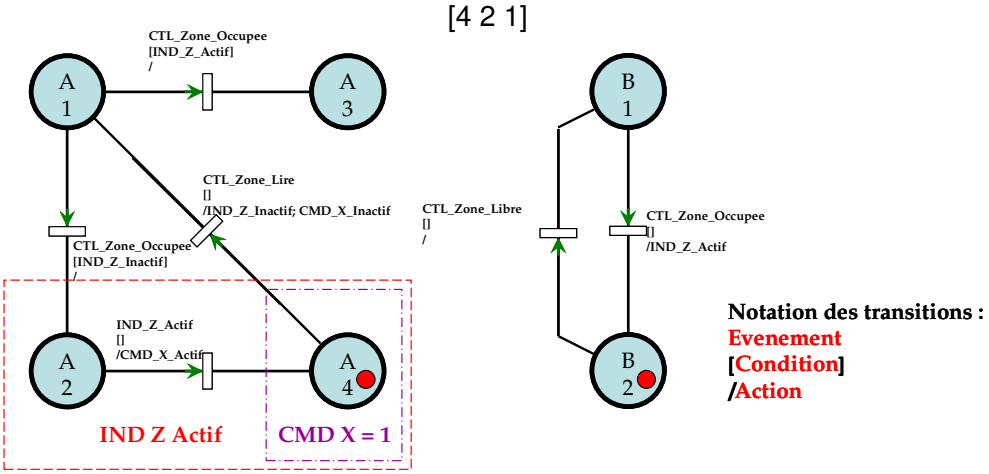
Beschreibung der Systemtransitionen
Externes Ereignis
[Zustandsvektor]

## C.2 Exemple 2 écrit en langage AEFD

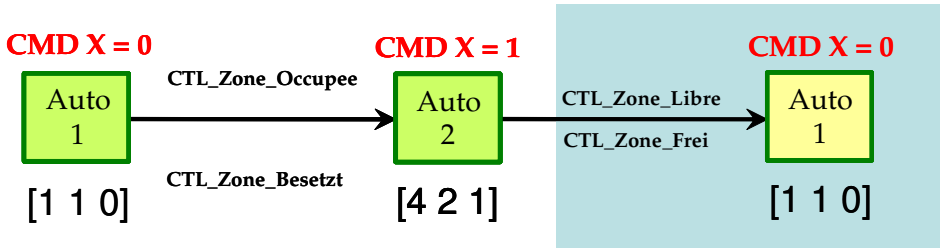
### C.2.1 – État initial du système




### C.2.2 – États fonctionnels du système




### C.2.3 – Arbre des états système




*Notation des transitions système:*
*Événement externe* = Externes Ereignis
[ Vecteur d'état = Zustandsvektor ]
Sortie / Ausgang : CMD X = 0 oder 1

Beschreibung der Systemtransitionen
Externes Ereignis
[Zustandsvektor]

*État existant précédemment*
Vorher existierender Zustand
*Vorher schon existierender Zustand*

## C.2 Beispiel zwei, in AEFD-Sprache geschrieben

### C.2.1 – Anfangszustand des Systems

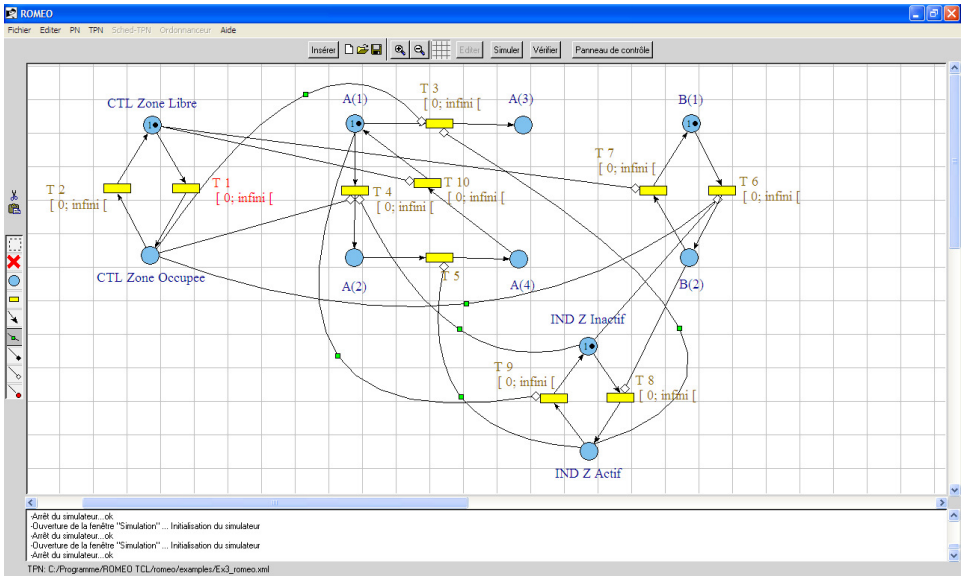
### C.2.2 – Funktioneller Systemzustand

### C.2.3 – Zustandsbaum des Systems

## C.3 Exemple 2 écrit en langage réseau de Petri (Roméo)

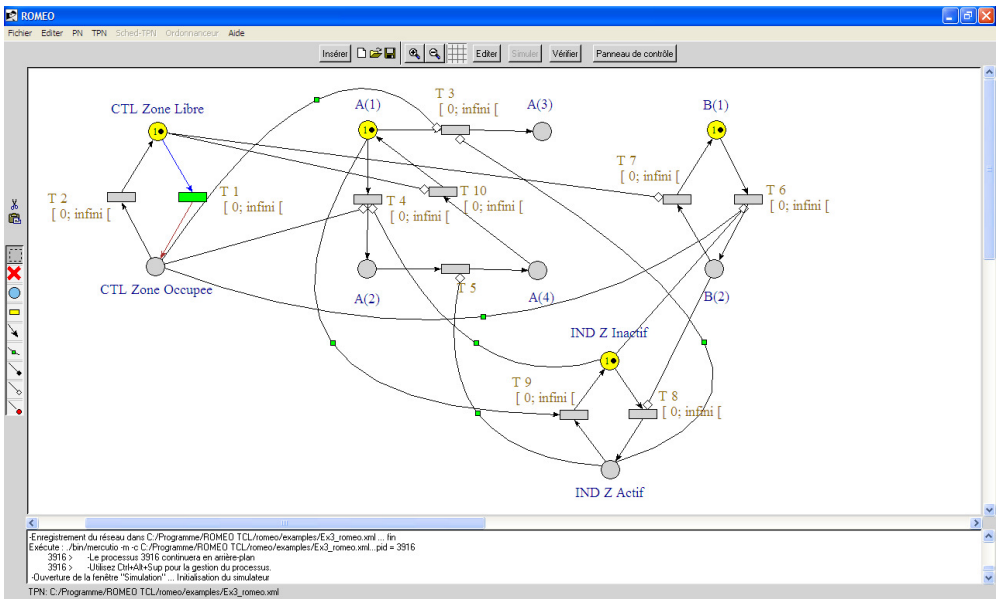
[Gardey, 2005]

### C.3.1 – État initial du système




### C.3.2 – États fonctionnels du système

Init




Le nombre d'états système accessibles est très supérieur à cinq car l'interprétation ne distingue pas les événements internes et externes.

## C.2 Beispiel zwei, als Petrinetz geschrieben (Romeo)

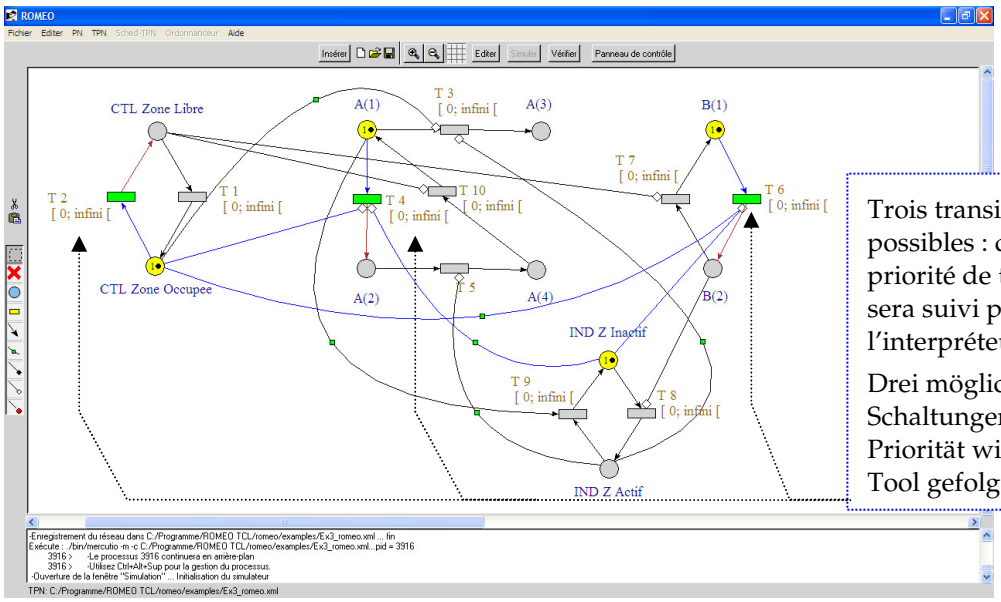
[Gardey, 2005]

### C.3.1 – Anfangszustand des Systems

### C.3.2 – Funktioneller Systemszustand

Die Anzahl der erreichbaren Systemzustände liegt weit über fünf, da der Interpreter nicht zwischen internen und externen Ereignissen unterscheidet.

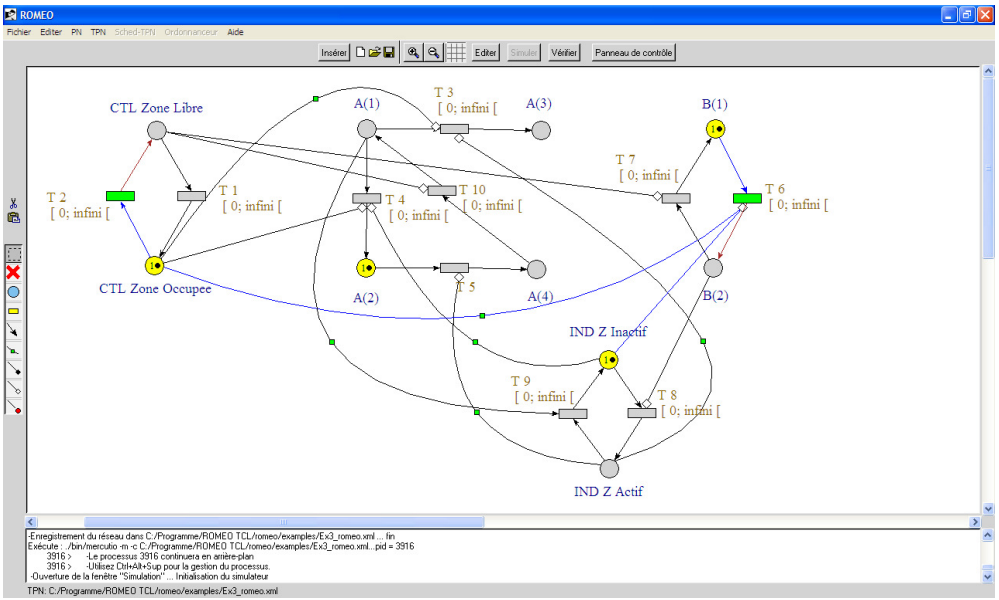
T1



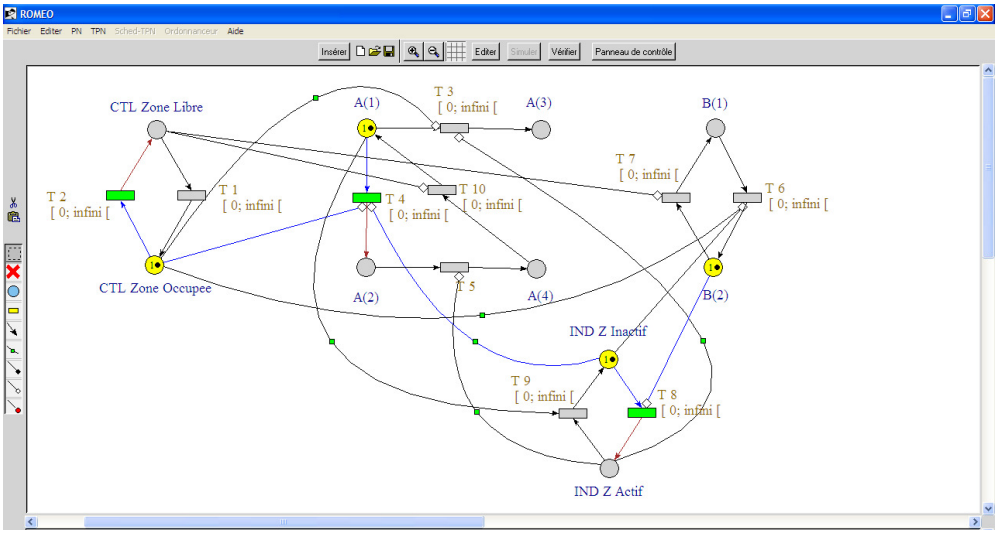

Trois transitions possibles : quelle priorité de tirage sera suivi par l'interpréteur ?

Drei möglichen Schaltungen: Welche Priorität wird vom Tool gefolgt?

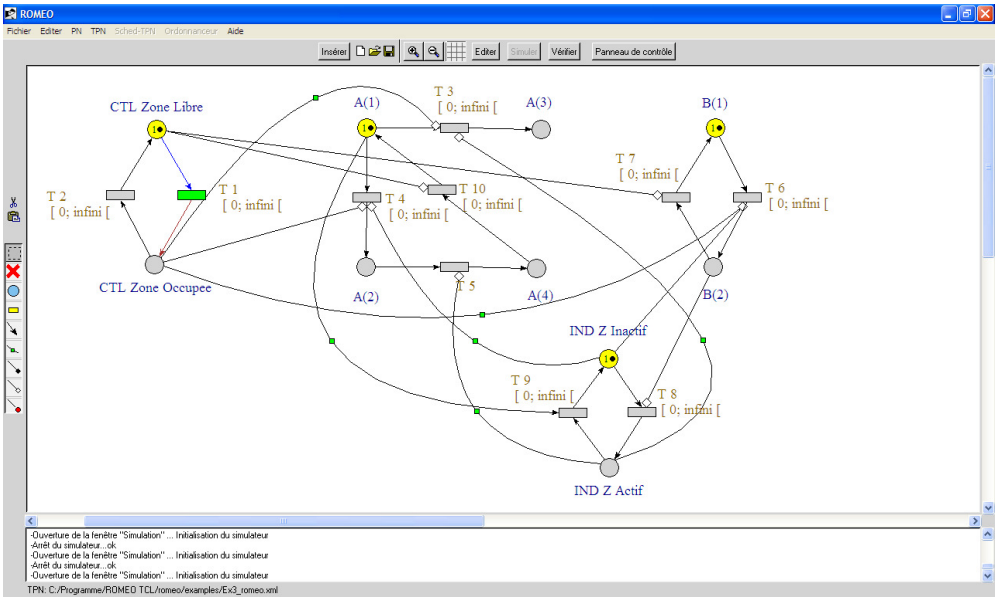
T4




T6



T2

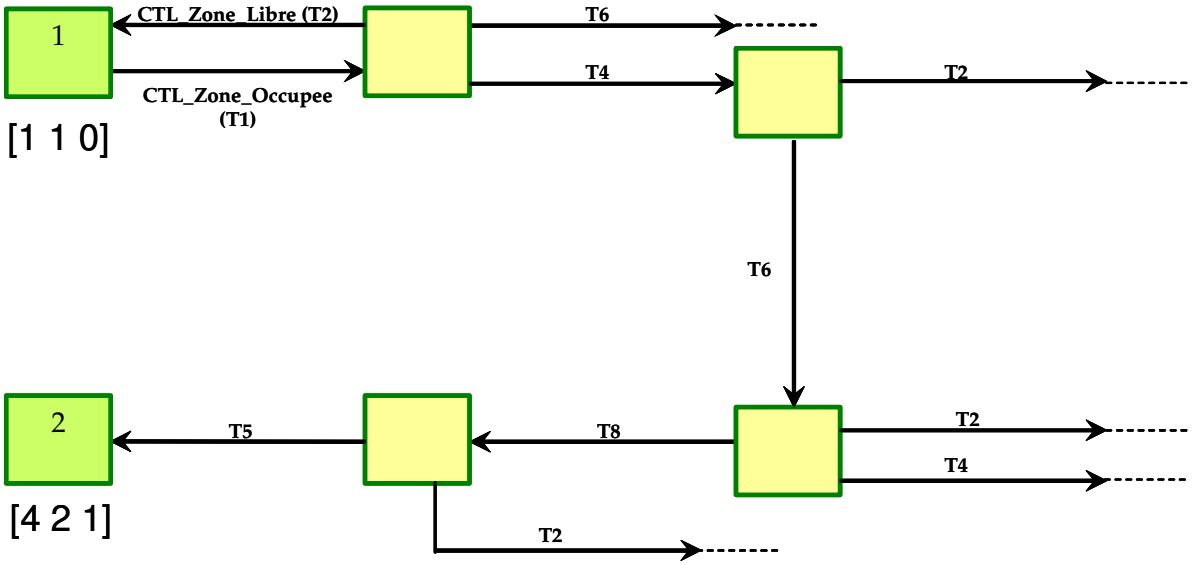



## C.3.3 – Arbre des états système

Le nombre d'états système accessibles est très supérieur à 5 car l'interprétation ne distingue pas les événements internes et externes

## C.3.3 – Zustandsbaum des Systems

Die Anzahl der erreichbaren Systemzustände liegt weit über fünf, da der Interpreter nicht zwischen internen und externen Ereignissen unterscheidet.




On montre donc qu'une description du logiciel d'application donné, peut donner lieu lors de l'exploration à soit une explosion combinatoire rapide dans le cas général de l'écriture sous forme de réseau de Petri, soit une maîtrise de la combinatoire des états système. Les RdP introduisent une complexité supplémentaire à la seule vision du spécificateur.

Es wurde also gezeigt, dass eine Formulierung einer bestimmten Anwendungssoftware bei der Auswertung entweder zu einer raschen kombinatorischen Explosion führen kann, vor allem bei der Beschreibung mittels Petrinetzen, oder zu einem Beherrschen der Systemzustandskombinationen. Petrinetze führen zu einer zusätzlichen Komplexität im Vergleich zur Sichtweise des Spezifizierens.

# Particularités et choix de traduction

# Besonderheiten und Übersetzungswahl

| Français | Deutsch |
|---|---|
| Actif (*état d'un indicateur*) | Aktiv (*Zustand eines Indikators*) |
| Action | Handlung |
| Analyse préliminaire des risques système (APRS) | Risikoanalyse des Systems |
| Arbre des états accessibles | Baum der zugänglichen Zustände |
| Arbre des états système | Zustandsbaum des Systems |
| Arc | Kanten |
| Automate (*ensemble de graphes réalisant les fonctions attendues de l'automate*) | Automat (*Gesamtheit der die Funktionen realisierenden Graphen, vom Automaten erwartet*) |
| Automate à nombre fini d'état ou automates à états finis (*automate dont le nombre d'états système est fini, dénombrable*) | Beendete Automat oder endliche Automaten (*ein Automat, dessen Systemzustände zählbar sind*) |
| Automate concurrentiel à contraintes (ACC) | Automat mit konkurrierenden Zwängen |
| Automate de preuve (*ensemble de graphes décrivant, pour une preuve et un automate fonctionnel, les propriétés devant toujours être vérifiées par l'automate*) | Beweisautomat (*Gesamtheit der Graphen, die für einen Beweis und einen funktionellen Automaten geschrieben sind, beschreiben die Automaten Eigenschaften, die immer überprüft werden müssen*) |
| Automates fonctionnels ou fonctionnel (*ensemble de graphes décrivant pour un poste d'aiguillage donné, les fonctionnalités ou fonctions à réaliser par l'automate – cet automate est un automate concurrentiel à contrainte et un automate à nombre fini d'états*) | Funktionelle Automaten (*Gesamtheit der für ein gegebenes Stellwerk beschreibenden Graphen, die Funktionalitäten oder die Funktionen, vom Automaten zu realisieren - dieser Automat ist ein Automat mit konkurrierenden Zwängen und ein Automat mit endlicher Zustandszahl*) |
| Automatisme à relais (*fonctions séquentielles réalisées à l'aide de relais de sécurité – fonction dite de sécurité*) | Relaisautomatismus (*sequenzielle Funktionen, die mit Hilfe von Sicherheitsrelais realisiert - vereinbarte Sicherheitsfunktionen*) |
| Changement d'état de l'environnement (*se traduit par un événement externe - changement d'état d'un équipement terrain, message, échéance d'une temporisation...*) | Änderung des Umgebungszustandes (*wird in Form eines externen Ereignisses ausgedrückt - Änderung des Zustandes einer Anlage vor Ort, Nachricht, Ende einer Verzögerung*) |
| Condition (*équation logique devant être vraie pour que l'événement associé soit pris en compte*) | Bedingung (*logische Gleichung, muss richtig sein, damit das vereinigte Ereignis berücksichtigt wird*) |
| Contrôle / Contrôleur d'aiguille | Überwachung / Weichenprüfer |
| Défaillances ou incidents | Störungen |
| Défaillances systématiques (*défauts systématiques*) | Systematische Fehler (*deterministische Fehler*) |
| Défaut (cause de la panne ou défaillance) | Defekt |
| Défauts | Fehler |
| Défauts de logiciels applicatifs | Fehler in der Anwendungssoftware |
| Durées de vie d'un système informatique :<br>- durée de modifiabilité du fonctionnel<br>- durée de réparabilité du hardware et du software système (mises à jours…)<br>- durée de fonctionnement sans modification | Lebensdauer einer IT-Systems:<br>- Dauer in der es Funktionsänderungsmöglichkeiten gibt<br>- Dauer in der es Möglichkeiten der Reparatur gibt (*Hard- und Grundsoftware des IT-Systems*)<br>- Benützungsdauer des IT-Systems |
| Élaboration des propriétés de sécurité du poste d'aiguillage (*il s'agit de la liste exhaustive des incompatibilités entre positions de ressources du poste d'aiguillage devant exister pour garantir la sécurité des circulations*) | Bestimmung der Sicherheitseigenschaften des Stellwerks (*es handelt sich um die erschöpfende Liste der Inkompatibilitäten der Stellmöglichkeiten des Stellwerkes, um die Sicherheit des Verkehrs zu garantieren*) |
| Entrée externe | Externes Ereignis |

| | |
|---|---|
| Erreur de spécification (*erreur du logiciel fonctionnel*) | Spezifikationsfehler (*Fehler der funktionellen Software*) |
| Essais de validation (manuels) | Validation |
| État non sûr du système | Unsicherer Systemzustand |
| État sûr du système | Sicheren Systemzustand |
| État système « sûr » et « non atteignable » (*depuis l'état initial : Surabondant*) | Systemzustand „sicher" und „nicht erreichbar" (*vom Anfangszustand aus : überflüssig*) |
| État système «non atteignable» (*depuis l'état initial du système*) | „Nicht erreichbarer" Systemzustand (*vom Anfangszustand des Systems ausgehend*) |
| État système «non sûr» et «atteignable» (*depuis l'état initial du système*) | „Unsicher" und „erreichbarer" Systemzustand (*vom Anfangszustand des Systems ausgehend*) |
| État système «sûr» et «déjà atteint» précédemment (*depuis l'état initial du système*) | Systemzustand „sicher" und früher „schon erreicht" (*vom Anfangszustand des Systems ausgehend*) |
| État système nouveau, «sûr» et «atteignable» (*depuis l'état initial du système*) | Neuer Systemzustand „sicher" und „erreichbar" (*vom Anfangszustand des Systems ausgehend*) |
| États fonctionnels du système | Funktioneller Systemszustand |
| Évènement externe (*changement d'état d'une entrée logique de l'automate*) | Externes Ereignis (*Zustandsänderung eines logischen Einganges des Automaten*) |
| Événement interne (*changement d'état d'un indicateur interne à l'automate – communication entre deux graphes de l'automate*) | Internes Ereignis (*Änderung des Zustandes eines internen Indikators im Automaten - Kommunikation zwischen zwei Graphen des Automaten*) |
| Exhaustivité des tests | Vollständigkeit der Testfälle |
| Exploration | Auswertung |
| Exploration automatique et systématique des états accessibles | Automatische und systematische Untersuchung der erreichbaren Zustände |
| Fonction de contrainte | Beschränkungsfunktion |
| Fonction de sécurité (*fonctions réalisée avec un niveau de sécurité SIL4 ou plus*) | Sicherheitsfunktion (*Funktionen, mit einem Sicherheitsniveau SIL4 realisiert oder mehr*) |
| Fonctions de contrainte (*sont décrites sous la forme d'incompatibilités*) | Beschränkungen (*werden in Form von Unvereinbarkeiten beschrieben*) |
| Franchissable | Schaltfähig |
| Graphe | Graph |
| Inactif (*état d'un indicateur*) | Inaktiv (*Zustand eines Indikators*) |
| Incompatibilité (*se traduit par deux enclenchements qu'il conviendra de réaliser au moyen de la table d'enclenchement ou des graphes fonctionnels*) | Unvereinbarkeit (*wird durch zwei Verschlüsse umgesetzt, die man auf einem Stelltisch oder funktionelle Graphen unterbringen muss*) |
| Interpréteur de réseaux de Petri (RdP) (*généralement ceux-ci ne distinguent pas les événements internes et externes*) | Petrinetz (PN) Interpreter (*im Allgemeinen unterscheiden diese nicht die internen und externen Ereignisse)* |
| Langage AEFD | AEFD Sprache |
| Le nombre d'états système accessibles | Die Anzahl der erreichbaren Systemzustände |
| Libre (*état d'un indicateur ou d'une entrée*) | Frei (*Zustand eines Indikators oder eines Eingangs*) |
| Maintenabilité d'une application informatique (*est l'aptitude à pouvoir être modifiée et testée rapidement*) | Instandhaltbarkeit einer IT-Anwendung (*ist die Möglichkeit zur raschen Änderung und zum raschen Testen der Anwendung*) |
| Marque ou jeton | Kante |
| Modifiabilité des systèmes informatiques | Veränderungsmöglichkeiten bei IT-Systemen |
| Occupé (*état d'un indicateur ou d'une entrée*) | Belegt (*Zustand eines Indikators oder eines Eingangs*) |
| Panne, défaillance ou incident | Störung |

| | |
|---|---|
| Pannes aléatoires du matériel (*défauts stochastiques*) | Zufällige Störungen der Hardware (*stochastische Fehler, nicht deterministische Fehler*) |
| Place | Platz |
| Place d'origine | Ursprungs Platz |
| Place destination | Ziel Platz |
| Postulats de fonctionnement (*événements externes ou séquences d'événement à ne pas prendre en compte car ne pouvant jamais survenir - Par exemple, les circuits électriques interdisent la mise en position « vrai » des contrôles à gauche et à droite*) | Anforderungen oder Funktionspostulate *(externes Ereignisse oder Sequenz auf Ereignisse die sich niemals ereignen sollen - Zum Beispiel, verbieten die Stromkreise die gleichzeitige Stellung des rechten und des linken Weichenprüfers auf die Position „wahr")*<br>*Oder aucc im Sinne* „Funktionelle Grungprinzipien" |
| Propriétés de surabondance (*spécifications conduisant à une situation indisponible et immédiatement sûre, mais qui peut conduire à une situation dangereuse du fait d'une erreur humaine dans l'application de procédure - Les incompatibilités assurant la sécurité ne doivent pas entraver les fonctionnalités attendues - L'existence d'enclenchements superflus peut être source d'insécurité par application de procédures manuelles règlementaires par les opérateurs*) | „Überflüssige" Eigenschaft oder „überflüssige" Forderungen *(Anforderungen, die sich ggf. widersprechen und die deswegen zu einem sicheren, aber ungewüschten Zustand führen, der unter unvorhersehbaren Bedingungen zu einem unsicheren Zustand werden kann. Anforderung, die auf ziemlich paradoxe Weise den Gesamtsicherheitslevel reduzieren können - Die die Sicherheit gewährleistenden Unverträglichkeiten dürfen die erwarteten Funktionen nicht beeinträchtigen - Das Bestehen von störenden oder überflüssigen Sicherungen kann zu Unsicherheit führen durch die Anwendung von manuellen Vorschriften durch das Bedienungspersonal)* |
| Propriétés de sécurité (*spécifications qui lorsqu'elles ne sont pas respectées, conduisent à une situation immédiatement dangereuse ou non sûre)* | Sicherheitseigenschaften (*Anforderungen, die sich ggf. wiedersprechen und die deswegen zu einem unsicheren und ungewünschten Zustand führen*) |
| Propriétés de sécurité à vérifier (*indépendamment de la manière dont le programme fonctionnel a été réalisé, elles peuvent être directement déduites des incompatibilités recherchées*) | Die zu prüfenden Sicherheitseigenschaften (*unabhängig von der Art, in der das funktionelle Programm verwirklicht worden ist - Sie können direkt von den gesuchten Unvereinbarkeiten abgeleitet werden*) |
| Réseaux de Petri interprétables | Interpretierbare Petrinetze (*Anwendungssoftware für die Definition der Sicherungen*) |
| Sortie externe | Externer Ausgang |
| Spécifications fonctionnelles | Funktionelle Spezifikationen |
| Système (*se dit de l'automate et de son environnement et automatismes à relais*) | System (*ein Automat und seiner Umwelt, Relaisautomatismen…*) |
| Système de sécurité | Sicherungssystem (≠Sicherheitssystem) |
| Système informatique ou informatisé | IT System |
| Techniques numériques | Digitale Technik |
| Tests de validation d'un système | Versuche zur Validation eines Systems |
| Transition (d'un état système à un autre) | Schaltung |
| Transition (d'une place à une autre) | Transition |
| Transitions système | Systemtransitionen |
| Validation formelle (automatique) | Formale Verifikation oder formale Überprüfung |
| Vérification au sens général | Überprüfung |

**Termes ferroviaires** **Eisenbahn Vokabular**

| Français | Deutsch |
|---|---|
| Absence de conditions surabondantes (*Il s'agit de vérifier qu'il n'existe pas d'enclenchement superflu. Celui-ci nécessiterait l'application de procédures réglementaires et pourrait conduire à une situation dangereuse. Lorsque l'indicateur P7 à l'état vrai, il existe au moins une condition surabondante*) | Abwesenheit überflüssiger (überreichlicher) Bedingungen (*Es handelt sich darum, zu überprüfen, dass es kein überflüssiges Einschalten (Verschluss) gibt. Dieses erforderte die Anwendung vorschriftsmäßiger Verfahren und könnte zu einer gefährlichen Situation führen. Wenn der Indikator P7 im richtigen Zustand ist, gibt es mindestens eine überreichliche Bedingung*) |
| Agent circulation ou d'exploitant ferroviaire | Betriebspersonal |
| Aiguillage | Weiche |
| Aiguille (lame d'aiguille) | Weiche Zunge |
| Aiguilleur | Weichensteller |
| Annonce au passage à niveau (PN) (*graphes d'annonce*) | Zugvormeldung am BÜ (*Vormeldungsgraphen*) |
| Application des procédures par les opérateurs | Anwendung der Vorschriften durch das Bedienungs-personal |
| Assurance qualité du processus | Qualitätssicherung |
| Automates de preuve (*définis pour valider le modèle du poste d'aiguillage*) | Beweisautomaten (*die definiert werden, um das Stellwerksmodell zu validieren*) |
| Avertissement | Vorsignal |
| Barrières de passage à niveau (PN) | Schranken oder Barriere |
| Block absolu (*lorsque que tout franchissement d'un signal fermé impose la délivrance systématique d'une autorisation de l'opérateur sédentaire*) | Absoluter Block (*mit unbedingten Haltsignalen / wenn jedes Überfahren eines geschlossenen Signals systematisch eine Erlaubnis seitens des Weichenstellers erfordert*) |
| Block automatique lumineux (BAL) | Selbstblock mit Lichtsignale |
| Block non permissif ou absolu (*se dit d'un système d'espacement lorsque le mécanicien n'est pas autorisé à pénétrer de sa propre initiative dans un canton non libre – signal d'arrêt utilisé : carré*) | Blocksystem mit unbedingten Haltsignalen (*Blocksystems, wenn es dem Lockführer gestattet wird, auf eigenen Initiative in einen nicht freien Blockabschnitt einzudringen - Benutztenhaltsignal: Carré*) |
| Block ou Système de block | Blocksystem |
| Block permissif (*le mécanicien d'un train suiveur peut pénétrer sous condition de vitesse (marche à vue) de sa propre initiative dans un canton non libre – signal d'arrêt utilisé : sémaphore ou feu rouge clignotant*) | Permissiver Block oder Blocksystem mit bedingten Haltsignalen (*der Lokführer darf über das Blocksignal fahren – Semaphor oder blinkendes rotes Licht*) |
| Boutons d'itinéraires (*pour postes PRS ou PRG*) | Fahrstraßenknöpfe (*für PRS und PRG Stellwerke*) |
| Circuit de voie (*circuit électrique utilisant les deux files de rail afin d'assurer sur une portion de voie les fonctions de sécurité : détection d'une part, l'absence de circulation et, d'autre part, l'absence de rupture de rail*) | Gleisstromkreis (*Stromkreis durch zwei Schienenstränge, um auf einem Gleisabschnitt die Sicherheitsfunktionen umzusetzen: Einerseits Fest-stellung, der Abwesenheit von Verkehr und, andererseits, die Abwesenheit eines Schienenbruches*) |
| Circuits électriques | Stromkreise oder elektrische Schaltkreise |
| Circulations ferroviaires | Fahrbetrieb |
| Commande à distance de postes d'aiguillage | Fernsteuerung von Stellwerken |
| Compteur d'essieux (*comptage assuré à ±0 en France, et ±1 en Allemagne*) | Achszähler (Berechnung mit *±0 in Frankreich, ±1 in Deutschland*) |
| Conditions surabondantes | Überflüssige Bedingungen |
| Conducteur | Lokführer |

| | |
|---|---|
| Contrôle impératif d'une aiguille pour la direction de droite (KAg D) | Prüfung die korrekte Lage der beweglichen Teile der Weiche für die rechte Richtung |
| Contrôle impératif d'une aiguille pour la direction de gauche (KAg G) | Prüfung die korrekte Lage der beweglichen Teile der Weiche für die rechte Richtung |
| Dérives | Entlaufen (*Das Wegrollen ist eine nicht beherrschte Fahrzeugbewegung*) |
| Détecteur électromécanique | Mechanische Gleisschaltmittel |
| Disponibilité | Verfügbarkeit |
| disponibilité opérationnelle du système | Betriebliche Verfügbarkeit des Systems |
| Dispositif d'annonce des circulations aux chantiers | Zugwarnanlage für Baustellen |
| En service – situation non sure | In Betrieb – unsichere Situation |
| En service – système totalement actif | In Betrieb – System aktiv |
| En service avec une défaillance détectée | In Betrieb – mit einer detektierten Panne |
| Enclenchement (*L'enclenchement est réciproque et interdit certaines combinaisons de la position des leviers, donc des aiguilles et signaux*) | Einschränkung oder Verschluss oder Sicherung (*Der Verschluss ist wechselseitig und verhindert bestimmte Stellkombinationen der Hebel, also der Weichen und der Signale*) |
| Enclenchement d'approche ou Zone d'approche (ZAp) | Annäherungsverschluss |
| Enclenchement de transit ou Transit souple (*fonction logique assurant l'enclenchement des aiguilles et des itinéraires : correspond à une zone orientée active à la formation d'un itinéraire jusqu'à son franchissement et sa libération par la circulation intéressée) (Le train libère une à une les aiguilles : libération progressive des aiguilles dégagées de l'itinéraire*) | Transit Einschränkung oder Fahrstraße mit Teilauflösung (*Der Zug befreit jede Weiche einzeln: nach und nach werden die sich in der teilaufgelösten Fahrstraße befindlichen Weichen freigemacht. Die Weichen sind frei wenn ihre Gleisstromkreise überfahren und freigemacht sind*) |
| Enclenchements | Verschlüsse |
| Enclenchements mécaniques | mechanischen Verschlüsse |
| Entrebâillement | Zungenklaffen |
| Entrées et sorties NS1 | NS1 Ein- und Ausgänge |
| Entrées terrains (CTL) | Eingänge (CTL) – externe Eingänge |
| Entreprises ferroviaires (EF) | Eisenbahnunternehmen |
| Erreur des opérateurs dans l'application des procédures | Fehler des Betriebspersonals bei der Anwendung der Vorschriften (Behandlung der Störung) |
| Essais réels sur machine cible | tatsächliche Versuche auf der Zielmaschine |
| États accessibles du système explorés | Erreichbare Systemzustände |
| Experts signalisation | Signaltechnikexperten |
| Exploitant | Betriebspersonal |
| Facteurs humains | Menschliche Faktoren |
| Fermeture automatique du signal de protection ou de block (*nécessaire car en France le block est permissif*) | Automatische Schließung des Deckungs- oder Blocksignals (*dies ist nötig da in Frankreich Block mit bedingten Haltsignalen*) |
| Fins de temporisation (FTP) | Ende der Verzögerung (FTP) |
| Fonction de sécurité (*définie pour un contexte d'usage et d'environnement*) | Sicherheitsfunktion (definiert *durch die Art der Verwendung und das Umfeld*) |

| | |
|---|---|
| Fonction de sécurité intrinsèque (ou fail safe) (*La sécurité intrinsèque est donc étroitement liée au principe déterministe. Mais elle laisse aussi, de fait, l'homme entièrement responsable de la sécurité. L'automatisme en défaut «passe la main» à l'homme*) | Systemintegrierten Sicherheit (*Die systemintegrierte Sicherheit ist stark mit dem deterministischen Prinzip verbunden. Aber sie lässt den Menschen allein für die Sicherheit verantwortlich. Eine gestörte Regelung übergibt ihre Aufgabe an den Menschen*) |
| Fonctionnel applicatif du poste ou fonctionnel (*écrits avec des graphes fonctionnels en langage AEFD*) | Stellwerksfunktionen (*mit funktionellen Graphen in AEFD Sprache geschrieben*) |
| Fonctions d'enclenchement | Sicherungsfunktionen |
| Fonctions d'enclenchement attachées à l'itinéraire | Fahrstraße mit verbundenen Verschlussfunktionen |
| Hors service ou en travaux | Außer Betrieb oder Baustelle |
| Incompatibilités | Unvereinbarkeit |
| Incompatibilités (*incompatibilité qui se traduit par deux enclenchements qu'il convient de réaliser*) | Unvereinbarkeiten (*wird durch zwei Verschlüsse umgesetzt*) |
| Indicateurs du fonctionnel (IND) | Indikatoren (IND) der Funktion |
| Indication « carré violet » | Violettes Licht (*Deckungsignale für Rangierzüge*) |
| Indication « carré » | Rotes Licht – Hauptsignal |
| Infrastructure | Infrastruktur |
| Installations | Einrichtungen – Anlagen |
| installation permanente de contre sens (IPCS) | ständigen Anlage zum Befahrenen des Gegengleises |
| Itinéraire | Fahrstraße |
| Itinéraire avec transit souple | Fahrstraße mit Teilauflösung gesteuert |
| Itinéraire établi | Fahrstraße gebildet (Eingestelltefahrstraße) |
| Langage AEFD | AEFD-Sprache |
| Le moteur de résolution est purement événementiel (*c'est à dire qu'une transition ne peut être franchie que sur présentation d'un événement*) | Die Lösungsmaschine ist rein ereignisbezogen (d. *h. ein Übergang ist nur bei Vorhandensein eines Ereignisses möglich*) |
| Le premier automatisme totalement informatisé | Die erste vollständig rechnergestützte Steuerung |
| Les commandes centralisées | Zentrale Steuerung |
| Levier d'aiguille | Weichenhebel |
| Levier de signal | Signalhebel |
| Leviers de parcours | Fahrstraßenhebel |
| Levier en position normale ou renversée | Hebel in normaler oder umgestellter Position |
| Maintenance des installations fixes | Instandhaltung der Infrastruktureinrichtungen |
| Maintenance des matériels roulants | Wartung des Fuhrparks |
| Matériel roulant | Fahrzeuge – Fuhrpark |
| Messages (MSG) | Nachrichten (MSG) |
| Monde ferroviaire clos | Abgeschlossener Bahnbereich |
| Moteur de résolution des graphes du fonctionnel | Lösungsmaschine der funktionellen Graphen |
| Nez à nez | Frontalzusammenstoß |
| Niveau de sécurité | Sicherheitsniveau |
| Organes de commande *(boutons, commutateurs…)* | Steuerorgane (*Knöpfe, Schalter…*) |
| Passage à niveau informatique | Elektronischer Bahnübergang |
| Passage à niveau à signalisation automatique à deux demi barrières (SAL2) | Bahnübergangs mit automatischer Signalisierung und zwei halben Barrieren (SAL2) |
| Panneaux lumineux | Lichtsignale |
| Plan de voie | Gleisplans |

| | |
|---|---|
| Poste à relais | Relaisstellwerk |
| Poste à Relais et à Commande Informatique (PRCI) | Rechnergesteuertes Relaisstellwerk (PRCI) |
| Poste Informatique de Nouvelle Génération (PAI NG) | Rechnerstellwerk neuer Generation (PAI NG) |
| Poste mécanique | Mechanisches Stellwerk |
| Poste tout Relais à transit Souple (PRS) | Stellwerk mit Fahrstraßenteilauflösung (PRS) |
| Poste tout Relais Géographique (PRG) | Geographisches Relaisstellwerk (PRG) |
| Postes d'aiguillage à itinéraires | Fahrstraßenstellwerken |
| Postes d'aiguillage Informatiques (PAI) | Rechnerbasiertes Stellwerk (PAI) oder Rechner-stellwerke |
| Postes électriques à leviers d'itinéraires | Elektrische Fahrstraßenstellwerke |
| Postes électromécaniques | Elektromechanische Stellwerke |
| Postes tout Relais à transit Souple (PRS) | Stellwerke mit Fahrstraßenteilauflösung (PRS) |
| Postulats de fonctionnements | Funktionsanforderungen |
| Principe d'application de la méthode preuve | Anwendungsprinzip der formalen Methode |
| Prise en écharpe | Flankenfahrt |
| Probabilités de dysfonctionnement contraires à la sécurité (*certains de ces dysfonctionnements réputés «sûrs» nécessitent de pouvoir compter sur les opérateurs dans le cadre l'application de procédure*) | Die Wahrscheinlichkeit einer sicherheitswidrigen Fehlfunktion (*bei einigen dieser „sicheren" Fehlfunktionen ist man bei der Anwendung der Vorschriften auf das Bedienpersonal angewiesen)* |
| Procédure de mise en service d'un poste d'aiguillage | Vorgehen bei Inbetriebnahme eines Stellwerks |
| Procédures | Vorschriften oder Betriebsregeln |
| Programme du poste | Stellwerksprogramm |
| Programme fonctionnel attendu du poste | vom Stellwerk erwartetes funktionelles Programms |
| FIFO | FIFO-Speicher |
| Rattrapage | Auffahren |
| Réglementation d'exploitation ferroviaire | Eisenbahn Forchriften |
| Régulateur | Betriebsfahrdienstleiter |
| Relais de sécurité | Sicherungsrelais |
| Relais électromécanique | Elektromechanischen Relais |
| Réseaux de Petri (*sont des graphes AEFD fonctionnels décrits dans un fichier ASCII et interprétés en temps réel par l'interpréteur de la machine cible*) | Petrinetze (*sind funktionelle AEFD-Graphen, die in einer ASCII-Datei definiert und vom Interpreter der Zielmaschine in Echtzeit interpretiert werden*) |
| Sécurité ferroviaire | Sicherheit bei der Eisenbahn |
| Sécurité intrinsèque | Inhärente Sicherheit (*basiert auf Hardwaresystemen, die mit spezifischer Sicherheit konzipiert wurden*) |
| Signal Carré : C (*commande l'arrêt avant le signal, n'est franchissable que sur ordre écrit*) | Rotes Hauptsignal: C (*Absolutes Haltsignal. Darf nur nach schriftlicher Anweisung überfahren werden*) |
| Signal Carré violet : Cv (*commande l'arrêt avant le signal sur voie de service, n'est franchissable que sur ordre écrit*) | Violettes Hauptsignal: Cv (*Absoluter Halt vor dem Signal auf Rangier- und Dienstgleisen*) |
| Signal d'avertissement : A (*commande d'être en mesure de s'arrêter avant le signal suivant*) | Vorsignal: A (*Befehl, in der Lage zu sein, vor dem nächsten Signal anzuhalten. Entspricht dem deutschen Vorsignal „Halt erwarten"*) |
| Signal de block | Blocksignal (*wenn ein Zug im Blockabschnitt ist*) |

| | |
|---|---|
| Signal de protection (*signal portant l'indication carré ou carré violet*) | Deckungssignal (Signal mit rotem oder violettem Licht) |
| Signal de ralentissement 30 : R (*commande de ne pas dépasser 30km/h au franchissement des aiguilles en aval du rappel de ralentissement 30 suivant*) | Langsamfahrt 30 Signal: R (*Befehl, die nächste direkt nach dem Erinnerungssignal liegende Weiche mit höchstens 30 km/h zu überfahren*) |
| Signal de rappel de Ralentissement 30 : RR (*commande de ne pas dépasser 30km/h au franchissement des aiguilles en aval*) | Erinnerung der Langsamfahrt 30 Signal: RR (*Befehl, die nachgelagerte Weiche nur mit 30 km/h zu überfahren*) |
| Signal Sémaphore : S - commande l'arrêt avant le signal, la plaque d'identification définit les possibilités de franchissement | Semaphor (Blockhalt): S - Befehl, vor dem Signal anzuhalten. Eventuelle Schilder geben an, ob das Signal auf Sicht überfahren werden kann |
| Signalisation | Signalsystem |
| Signaux de cantonnement | Blocksignale |
| Signaux de limitation de vitesse | Langsamfahrsignale |
| Systèmes de contrôle commande qui assurent des fonctions de sécurité | Systeme zur Zugsteuerung und -sicherung |
| Système ferroviaire | Bahnsystem |
| Système Normalisé de Télétransmissions Informatiques (SNTI) | genormtes System mit Datenfernübertragung (SNTI) |
| Table d'enclenchement | Stelltisch |
| Tableau de contrôle optique (TCO) | Gleistafel - Überwachungsanzeige (TCO) |
| Tableau indicateur de vitesse (TIV) sur pointe mobile (*Annonce d'une zone de limitation de vitesse relative au franchissement en voie déviée d'un appareil de voie*) | Bewegliche Anzeige für die Ankündigung einer Geschwindigkeitsbegrenzung für die Zungenspitzen (*Ankündigung einer Geschwindigkeit, die beim Abbiegen auf der nächsten Weiche einzuhalten ist)* |
| Taux de mise en défaut | Fehlerrate |
| Temporisation de libération (*autorise le réarmement d'une annonce au passage à niveau lorsque toutes les conditions sont libres depuis plus de soixante secondes*) | Verzögerung der Auflösung (*gestattet die Wiederaufrüstung einer Zugvormeldung für einen Bahnübergang, wenn alle Bedingungen seit mehr als sechzig Sekunden frei sind*) |
| Test d'intégration | Integrationstests |
| Trou d'enclenchement | Sicherungsloch |
| Usage contrôlé | Überwachte Betrieb |
| Vecteur d'état | Zustandsvektor |
| Verrouillage d'une lame d'aiguille (*se dit d'une lame d'aiguille immobilisée par un dispositif solidarisant la lame du contre aiguille perpendiculairement à l'effort – n'existe pas en Allemagne car les aiguillage doivent rester talonnables*) | Verschluss (Verriegelung) einer Weichenzunge (*sagt sich einer Zunge von einer Anlage blockierter, die Zunge tritt entgegen, Weiche im rechten Winkel zur Anstrengung - existiere in Deutschland nicht, weil Weiche sollen von einem ihm vom Heck herangehenden Zug umgeworfen werden können)* |
| Voie unique | Eingleisigen Strecke |

# Bibliographie

# Literatur

[01Info, 2001] *Les méthodes formelles, garantes de la sécurité des systèmes* – « 01 Informatique » Revue n°1646 – 09/2001

[01Info, 2002] *Les méthodes formelles sonnent le glas des tests unitaires* – 01 Informatique n°1662 – 01/2002

[Abrial, 1996] *The B Book– Assigning Programs to Meanings* - J-R Abrial - Cambridge University Press - 1996

[Allain, 2006-1] *Formalization and Simulation of Operating Rules Using Colored Petri Nets* - L.Allain, O. Lahlou, P. Bon - Computers in Railway X, pp. 329-340, 2006.

[Amey, 2006-2] *Combining Model-driven Design with Diverse Formal Verification* - P. Amey, B.Dion - Praxis HIS, 20 Manvers St. Bath, BA1 1PX, UK - ERTS 2006 Toulouse – 01/2006

[Antoni, 2005] *Modular Line Management System – SYMEL* – M.Antoni, F. van Deth – ASPECT05 – IRSE – London - 06/2005

[Antoni, 2006] *Conception d'automatismes informatiques de sécurité, pérennes et économiques* – M.Antoni - NTIC06 CNAM – 05/2006

[Antoni, 2007-1] *Feasibility Study for the Implementation of a Formal Proof of Interpretable Specification* - M. Antoni, N. Ammad, FORMS FORMAT 2007 – Proceedings of Formal Methods for Automation and Safety in Railway and Automotive Systems (G. Tarnai and E. Schnieder Eds.), Braunschweig - 2007.

[Antoni, 2007-2] *Formal Validation Method for Computerized Railway Interlocking Systems* - M.Antoni, N.Ammad – ICSSEA 2007 20th International Conference - Software & Systems Engineering and their Applications – CNAM - Paris, France - 12/2007

[Antoni, 2007-3] *Apports des réseaux de PETRI pour une validation formelle d'un fonctionnel ferroviaire* – M.Antoni, N.Ammad – Séminaire « Ingénierie des système complexes à logiciels prépondérants » de la DGA - Toulouse - 11/2007

[Antoni, 2008-1] *The Ageing of Signalling Equipment – The Impact on Maintenance Strategies* – M.Antoni - RCM08 – 4thIET – Railway Condition Monitoring - 06/2008

[Antoni, 2008-2] *Une méthode de maîtrise des risques lies aux systèmes informatiques complexes et surs : la validation formelle* – M.Antoni – Université technologique de Troyes – 3SGS08 - Troyes 05/2008

[Antoni, 2008-3] *Formal Validation Method for Computerized Railway Interlocking Systems* – M.Antoni, N.Ammad - WCRR08 - Signalling II of Operations - 05/2008

[Antoni, 2008-4] *Une méthode de validation formelle des systèmes informatiques de sécurité* – M.Antoni, N.Ammad – Congrès Lambda mu 16 Avignon - 10/2008

[Antoni, 2008-5] *The ageing of signalling equipment and the impact on maintenance strategies* – M.Antoni - ASPECT08 – IRSE Conference - London – 09/2008

[Antoni, 2009-1] *Buchkapitel : Sécurisation des architectures informatiques – Exemples concrets" – „Conception d'un module d'enclenchement informatique"* – M.Antoni – Hermès Lavoisier – 06/2009

[Antoni, 2009-2] *Formal validation of computerized interlocking system* – M.Antoni – CIE39 International conference on Computers & Industrial Engineering - Troyes – 07/2009

[Antoni, 2009-3] *The ageing of signalling equipment : News developments* – M.Antoni – CIE39 International conference on Computers & Industrial Engineering - Troyes – 07/2009

[Antoni, 2009-4] *Conception d'un module d'enclenchement informatique, éléments de bases de postes d'aiguillage informatiques conçus par la SNCF* (Entwürft für ein Modul des computergesteuerten Verschlusses, Basisbehandelteil der von der SNCF gebauten Rechnerstellwerke) – M.Antoni, N.Ammad – Revue RGCF n°185 – Edition Delville – 08/2009

[Antoni, 2009-5] *Formal Validation Method and Tools for French Computerized Railway Interlocking Systems* – M.Antoni – International Journal of Railway 2009.9 Vol.2 N°3 – 09/2009

[Arabestani, 1999-1] *Ein Weg zur Einsetzbarkeit formaler Methoden für Ingenieure im Eisenbahnwesen* – S.Arabestani, J-T.Gayen - FORMS99 –Workshop – Braunschweig – 12/1999

[Arabestani, 1999-2] *Modellierung des eingleisigen Bahnübergangs im Funkfahrbetrieb mit den Mitteln der UML* – S.Arabestani – FORMS99 -Workshop, Braunschweig – 12/1999.

[Arabestani, 2000] *Prinzip der Vererbung bei der objektorientierten Analyse am Beispiel der funkbasierten Bahnübergangssteuerung* – S.Arabestani, J-T.Gayen - FORMS2000 – Institut für Eisenbahnwesen und Verkehrssicherung IfEV, Technische Universität Braunschweig – 2000

[Barbier, 2009] *Le frisson d'Icare* – C. Barbier – L'Express – 06/2009

[Bahr, 2008] *Sicherprogrammierbare Steuerungen- Die Neuausrichtung in der Signaltechnik* – D.Bahr, R.Saykowski, J.Börcök, J.Hölzel – Signal +Draht – Rail Signalling and Telecommunication – 11/2008

[Barbu, 2007] *Basic Requirements for Use of Formal Tolls in Safety Procedures of Railway Systems* – G.Barbu – UIC IT & Satellite Navigation - FORMS07 – 2007

[Bastide, 2000] Habilitation der Universität von Toulouse : *Spécification comportementale par réseaux de PETRI : Application aux systèmes distribués à objets et aux systèmes interactifs* – R.Bastide – 01/2000

[Behm, 1999] Buch: *Meteor: A Successful Application of B in a Large Project*, FM'99, Toulouse, France, 1999 – P.Behm, P.Benoît, A.Faivre, J.Meynadier

[Belmonte, 2008] *Utilisation d'un modèle d'accident systémique comme référentiel commun à une analyse de risque interdisciplinaire* – F.Belmonte, J-L.Boulanger, W.Schön – CIFA 2008, Bucarest – 09/2008

[Berard, 2001] Buch: *Systems and Software Verification Model-Checking Techniques and Tools* – *B*.Berard, M.Bidoit, A.Finkel, F.Laroussinie, A.Petit, L.Petrucci, P.Schnoebelen - École Normale Supérieure de Cachan, France - 2001. XII, 196 p. 67

[Bernard, 1999] *Formale Methoden in der Praxis* - Österreichischer ISA-EUNET Workshop, Wien - Bernhard K. Aichernig und Peter Lucas - Ordinariat für Softwaretechnologie, TU-Graz – 4/1999

[Bertot, 2008] *Sémantique des langages de programmation première partie: sémantique naturelle* – Y.Bertot – 02/2008

[Bielinski, 1993] *Méthode de validation formelle* – P.Bielinski - Thèse soutenue en 1993 à l'Université Paris 6 - P.Bielinski, I*mplantation VLSI d'un algorithme de code correcteur d'erreur et validation formelle de la réalisation* - 1993.

[Bied, 1998] *Sécurité intrinsèque et sécurité probabiliste dans les transports terrestres* – D.Bied-Charreton - Synthèse n°31 de l'INRETS - 1998

[Bied, 2003] *Informatique et sécurité ferroviaire (Informatik und Sicherheit bei der Eisenbahn)* – D.Bied-Charreton – Revue RGCF – 12/2003

[Bitsch, 1999] *Strukturierte Erstellung von Sicherheitsspezifikationen in UML mit Hilfe der FMEA-Methode* – F.Bitsch, E.Canver, A.Moik – FORMS1999 - Formale Techniken für die Eisenbahnsicherung, Hrsg. E. Schnieder, Friedemann -Berichte VDI, Reihe 12, Verkehrstechnik/Fahrzeugtechnik, Nr.436, VDI Verlag GmbH, Düsseldorf 2000, S. 225-245

[Bitsch, 2000] *Classification of Safety Requirements for Formal Verification of Software Models of Industrial Automation Systems* – F.Bitsch - In Proceedings of 13th International Conference on Software and Systems Engineering and their Applications (ICSSEA), CNAM – Paris, France - 2000

[Bitsch, 2002] *Software Engineering in der industriellen Praxis* - In Tagungsband VDI-Bericht-Nr. 1666, Düsseldorf: VDI Verlag GmbH, S. 29-40. 2002.

*Spezifikation von Sicherheitsanforderungen mit Safety-Patterns* – F.Bitsch, Prof. Dr.-Ing. Dr. h. c. Peter Göhner - Institut für Automatisierungs- und Softwaretechnik, Universität Stuttgart

[Bitsch, 2003] *A way for applicable formal specification of safety requirements by tool – Support* – F.Bitsch - FORMS03 – 03/2003Techniques with Applications in Engineering", 03/2000

[Bohn, 2002] *Modeling and Validating Train System Applications Using Statemate and Live Sequence Chart* – J.Bohn, W.Damm, H.Wittke, J.Klose - Integration of Modelling Techniques for Train Control Systems, Integrated Design and Process Technology, IDPT-2002, June, 2002

[Bombardier, 2005] *Analyse technique – Rame 105 – La Tour de Carol – Septembre 2005* – Rapport d'expertise Bombardier Transportation / Document interne SNCF – 10/2005

[Bon, 2006] *BRAIL : De UML à B pour la modélisation d'un passage à niveau* - P.Bon, G.Mariano - FORUM NTIC06 Systèmes & Logiciels pour les NTIC dans le Transport - INRETS/ESTAS/ENST – 05/2006

[Bonhomme, 2001] Doktorarbeit *Réseaux de PETRI P temporels : contributions à la commande robuste* – P.Bonhomme – Université de Savoie - 07/2001

[Boulanger, 1999] Buch *METEOR:Validation de Spécification par modèle formel* – J-L.Boulanger, V.Delebarre et S.Natkin - RTS, p.47-62 - 06/1999

[Boulanger, 2000] *Processus de validation basée sur la notion de propriété-* J-L.Boulanger, M.Gallardo - Lambda Mu 12 Montpellier - 03/2000.

[Boulanger, 2006] *From UML to B - A Level Crossing Case Study* - G.Marianom, J-L.Boulanger, P.Bon - Computers in Railway pp. 351-362 - 2006.

[Boulanger, 2007-1] *Validation des données liées à la sécurité* – J-L.Boulager– Heudiasyc - Université de Technologie de Compiègne, France

[Boulanger, 2007-2] *État de l'art de la validation des données dans le domaine ferroviaire* – J-L.Boulanger – Heudiasyc - Université de Technologie de Compiègne, France – REE n°3 – 03/2007

[Boulanger, 2008] *B-Rail : D'UML à la méthode B pour améliorer un passage à niveau (B-Rail : from UML to the B Method in order to Model a Level Crossing)* – J-L.Boulanger, P.Bon - Recherche transport sécurité 95 (2007) 141-173 – 2008

[Bouvarel, 2003] *Initiation à l'exploitation ferroviaire* – P.Bouvarel – Cours Master Ferroviaire à l'école nationale des ponts et chaussées (ENPC) – 03/2003

[Braband, 2007] *A Proposal for Common Safety Methods for Technical Systems in European Railways* - J.Braband – Siemens AG transportation systems - FORMS07 – Rail Automation – Braunschweig – 2007

[Braband, 2008] *Nachweis mindestens gleicher Sicherheit gegenüber Referenzsystemen"*- Jens Braband – Signal+Draht – Rail Signalling and Telecommunication – 12/2008

[Brendel, 2003] *Analysis, numerische Analysis und Numerik für eine Differentialgleichung aus der Bildverarbeitung* – A.Brendel – Diplomarbeit – Albert Ludwigs Universität Freiburg im Breisgau Mathematisches Institut Abteilung für Angewandte Mathematik - 05/2003

[Butler, 2002] Buch – M.Butler, L.Petre, and K.Sere (Eds.): IFM 2002, LNCS 2335, pp. 339–359, 2002.- Springer-Verlag Berlin Heidelberg 2002

[Cachan, 1999] Buch : *Pour tout savoir sur le Post**, le livre du labo. de Cachan - Ph.Schnoebelen et al. Vérification de logiciels : techniques et outils du model checking". Vuibert, 1999.

[CALIFE, 2001] *CALIFE - Environnement pour la preuve formelle et le test d'algorithmes utilisés en télécommunication - Sous-Projet 3 : Génération de séquences de test - Techniques de génération de séquences de test temporisées : description d'une implémentation* – R.Castanet, P.Félix, P.Laurençot, D.Rouillard - ver 0.1 - 09/2001

[Caron, 1997] Buch : *Histoire des chemins de fer en France – 1740 à 1883* - Édition Fayard - François Caron – 05/1997

[Cassir, 2007] *Overview of Current Status for Common Safety Targets and Common Safety Methods* - C. Cassir, T.Breyne, P.Mihm, R.Piazza, FORMS07 – European Railway Agency, Safety unit, Valenciennes – 01/2007

[CEI60880, 1986] CEI 60 880 : *Logiciel pour les calculateurs utilisés dans les systèmes de sûreté des centrales nucléaires* - 1986

[CFTL, 2004] *Glossaire CFTL/ISTQB des termes utilisés en tests de logiciels* - International Software Testing Qualification Board – 12/2004

[Chartier, 2003] *Évaluation de la SdF des logiciels de contrôle commande à la RATP - Bilan et perspectives* – P.Chartier - Atelier de Qualification des Logiciels 06/2003

[Clearsy, 2006] *Approche formelle pour la réalisation d'un système sécuritaire de contrôle commande de façades de quais* – Clearsy – 06/2006

[Clearsy, 2008] *Présentation B pour la conception de systèmes* - Clearsy - Aix-en-Provence - 2008

[Coen, 2003] *Méthodes formelles et semi formelles - A Formal Approach for Designing CORBA-Based Applications – A.Coen-Porisini, M.Pradella, M.Rossi, D.Mandrioli – MGL806 – ACM Transactions on Software Engineering and Methodology, Vol.12, No.2, 04/2003.*

[Daumas, 2008] *Analyse d'un logiciel de sûreté par modélisation exécutable et instrumentée de ses documents de spécification et de validation* – F.Daumas – Institut de Protection et de Sûreté Nucléaire Département Évaluation de Sûreté B.P.6 - 92265 Fontenay aux roses

[DB301, 2003] *Deutsche Bahn AG: GeMEInsames Signalbuch* – 301DS/DV – Frankfurt am Main – 2003

[DB408, 2004] *Deutsche Bahn AG: Züge fahren und Rangieren* – DV408 - Frankfurt am Main – 2004

[Descubes, 1898] *Etude sur les enclenchements – Introduction des enclenchements conditionnels dans les tableaux etdétermination de tous les enclenchements secondaires qui en sont la conséquence* – M.Descubes (Ingénieur en chef adjoint de la voie à la compagnie de l'est) - Revue RGCF (revue générale des chemins de fer et des tramways) – *11/1898*

[DDaily, 1998] *Software error caused Delta III explosion* – Defence daily – Academic journals & books au Questia On line library - 10/1998

[Defossey, 2007] *Formal methods and temporal safety requirements : a level crossing application* - F.Defossez, P.Bon, S.Collart Dutilleul, INRETS/ESTAS und LAGIS – FORMS07 – 01/2007

[Defossey, 2008] *Temporal Requirements Checking in a Safety Analysis of Railway Systems* – F.Defossez, S.Collart-Dutilleul, P.Bon – Format 2008 - FORMS08 – 10/2008

[Defossez, 2008] *Temporal Requirements Checking in a Safety Analysis of Railway Systems* - F.Defossez, S.Collart-Dutilleul, P.Bon – FORMS 2008 - 10/2008

[Desroche, 2003] Buch: *La gestion des risques – Principes et pratiques – Chapitre 7 : La gestion des risques informatiques* - A.Desroche, A.Leroy, F.Vallée – Hermès Lavoisier – 02/2003

[Delhaye, 1993] *Pilotage automatique d'hélicoptère – Des spécifications enfin validées, Real-Time Systems* – F.Delhaye, C.Joubert – 1993

[Dewez, 2008] *Spécification logique et validation des programmes séquentiels* - L.Dewez - CNAM - NFP120 – Sciences et technologies de l'information et de la communication - UE - 2008-2009

[DFG, 2000] *Project of the focus area program (1064) on the "Integration of Specification Techniques with Applications in Engineering"* - Progress Report of the DFG project SafeRail: Integration von Methoden zur Spezifikation und Verifikation von Sicherungseinrichtungen im spurgeführten Verkehr - 03/2000

[DI603, 1993] *Veröffentlichungen des Verkehrswissenschaftlichen Institutes der technischen Hochschule Aachen – Der Einfluss des menschlichen Fehlers auf die Sicherheit der Eisenbahn* (DI603) – Albrecht Hinzen – 1993

[Dill, 1989] *Timing Assumptions and Verification of Finite-State Concurrent Systems* - D.L.Dill. – Workshop Automatic Verification Methods for Finite-State Systems, volume 407, pages 197–212, 1989.

[DO178B, 1992] Norme ED-12B / DO 178-B : *Software Considerations in Airborne Systems and Equipment Certification* - version A, 1982, version B - 1992

[Draghici, 2008] *La modélisation et la simulation en vue de la conduite des systèmes de production* – G.Draghici, N.Brinzei, I.Filipas – Universitatea Politehnica in Timisoara – 2008

[Dutruit, 2007] *Une introduction aux méthodes de modélisation et d'évaluation des performance en sûreté de fonctionnement des système* – Y.Dutuit – Université Bordeaux 1 – IMS/LAPS/CNRS UMR 5131 – Séminaire « Ingénierie des système complexes à logiciels prépondérants » - Toulouse - 11/2007

[ED12B-DO178B] *Software Considerations in Airborne Systems and Equipment certification* - Ver B, 1992

[EHESC, 1973] *Histoire de l'exploitation d'un grand réseau : la Compagnie des chemins de fer du Nord"* - Paris, École des hautes études en sciences sociales - 1973

[Ehrig, 2004] *Integration of Software Specification Techniques for Applications: Priority Program SoftSpez of the German Research Foundation*" (DFG) - H.Ehrig, W.Damm, J.Desel, M. Große-Rhode, W. Reif, E. Schnieder, E. Westkämper, Hrsg. - Final Report. 628 S., Springer Verlag, Lecture Notes in Computer Science (LNCS) Nr. 3147, Berlin, ISBN 3-540-23135-8 - 2004

[EN50126, 2000] EN 50126 : *Applications ferroviaires – Spécification et démonstration de la fiabilité, de la disponibilité, de la maintenabilité et de la sécurité (FDMS) – Bahnanwendungen – Spezifikation und Nachweis der Zuverlässigkeit, Verfügbarkeit, Instandhaltbarkeit, Sicherheit (RAMS)* - 01/2000

[EN50128, 2001] EN 50128 : *Applications ferroviaires – Système de signalisation, de télécommunication et de traitement – Logiciels pour systèmes de commande et de protection ferroviaire* - 07/2001

[EN50129, 2003] EN 50129 : *Applications ferroviaires – Systèmes de signalisation, de télécommunication et de traitement – Systèmes électroniques de sécurité pour la signalisation – Railway Applications – Communication, Signalling and Processing Systems – Safety related Electronic Systems for Signallingg"* - 02/2003

[Esterel, 2004] *Formal Verification for Model-Based Development* – A.Bouali, B.Dion – SAE International 2004 – Esterel Technologies - 2004

[Esterel, 2008] *Die Zertifizierung bestätigt die neueste SCADE Suite™-Codegenerierungstechnologie für kritische Softwareapplikationen in Eisenbahn-, Industrie- und Automobianwendungen* - Esterel Technologies' KCG™-Codegenerator erhält Zertifizierung für SIL 3/4 unter EN 50128 und SIL 3 unter IEC 61508 – Elancourt, Frankreich und Boston, Massachusetts — 22/09/2008

[Esterel, 2008] *Qualified Code Generation - SCADE Suite KCG - EN 50128 Code Generator – EN 50128 Handbook abstract* – 10/2008

[Evangelista, 2006] *Méthodes et outils de vérification pour les réseaux de PETRI de haut niveau. (Outils QUASTAR) / Application à la vérification de programmes Ada Concurrents* – S.Evangelista - Centre de Recherche en Informatique du CNAM – Paris - Doktorarbeit 12/2006

[Faivre, 1999] *Safety Critical Software of Meteor Developed with the B Formal Method and the Vital Coded Processor"* – A.Faivre, P.Benoit - World Congress on Railway Research (WCRR) - 1999.

[FIF, 2008-1] *La détection des trains* - Cours FIF (Fédération des Industries Ferroviaires française) – I6_Poré_SIG1_3_1 – 09/2008

[FIF, 2008-2] *Maintenance des installations de signalisation* - Cours FIF (Fédération des Industries Ferroviaires française) – I7_Antoni_SIG8 –- 10/2008

[FIF, 2009-1] *Les enclenchements – Introduction* - Cours FIF (Fédération des Industries Ferroviaires française) – I6_Poré_SIG1_3_3 –- 01/2009

[FIF, 2009-2] *Dangers – Rôles de la signalisation ferroviaire* - Cours FIF (Fédération des Industries Ferroviaires française) – I6_Poré_SIG1_2 – 01/2009

[Frey, 1998] *Transparent SPS-Programmierung unter SFC : Keine Freiheit ohne Grenzen* – G.Frey – IEE Automatisierung + Datentechnik – Hüthig Verlag – S.14-15 – 1998

[Frey, 1999] *Integration von Petrinetzen in eine IEC 1131 Entwicklungsumgebung* – G.Frey – Lehrstuhl für Automatisierungstechnik, Fachbereich Elektrotechnik Universität Kaiserslautern – Tagungsband SPS/IPC/DRIVES99 Nürnberg - Hüthig Verlag, Heidelberg, S.118-127 – 1999

[Frey, 2002] *Design of Formal Analysis of Petri Net Based Logic Control Algorithms* - Dissertation Univ. Kaiserslautern in Berichte aus der Automatisierungstechnik Aachen – Shaker Verlag - 2002

[Frey, 2003] *Simulative Validierung Petrinetz-basierter Steuerungen durch Transformation in Modelica* – Proceedings 17.Symposium Simulationstechnik (ASIM 2003) – Magdeburg – S.81-86 – 09/2003

[Fribourg, 1994] *Automates concurrents à contraintes* – L.Fribourg, M.Veloso Peixoto - CNRS – 1994

[Gardey, 2005] *Romeo: A Tool for Time Petri Nets Analysis* - CAV 2005 – Edinburgh – G.Gardey, D.Lime, M.Magnin, O-H. Roux - 2005

[Gartner, 2008] *Datarequest Insight : Unplanned Downtime Rising for Mission-Critical Application* - ,Gartner research- 10/2008

[Gartner, 2009] *Worldwide Trends of Formal Methods Application and the Issues in Information Systems to secure software dependability* - Gartner Consulting Japan – March 2009

[GDRGPL, 2008] *B pour la sûreté des logiciels et systèmes complexes par construction* - D.Cansell et D.Bert - Arbeitsgruppe GDR GPL

[Georgelin, 1998] Buch : *Simulation symbolique et preuve de descriptions VHDL avec ACL2* – P.Georgelin – TIMA Laboratory, Grenoble, France – 1998

[Gernigon, 1998] Buch : *Histoire de la signalisation ferroviaire française* – A.Gernigon - La Vie du Rail et des Transports, 07/1998.

[Goeb, 2003] *Gefährdung durch Computereinsatz in der Industrie* - Proseminar: Zuverlässigkeits- und Sicherheitsaspekte in technischen Systemen – M.Goeb – 11/2003

[Gomez, 2001] *ERTMS Driving and Operation Simulator under Distributed Architecture in a Virtual Reality Environment* – A.Gómez-Rey, J.M. Mera - ITEC'2001. Lille, France – 04/2001

[Gouda, 2008] *Un Thalys heurte un autre train aux Pays-Bas, pas de victimes* – Communiqué SNCB – 11/2008

[Grenet, 1998] *Contraintes d'ordre et automates d'arbres pour les preuves de terminaison* – T.Grenet – Université Henri Poincaré, Nancy 1 - Thèse 09/1998

[Grude, 1988] *Petrinetze eine informelle Einführung in die Grundideen* – U.Grude - 1988

[Guarnieri, 2008] *De l'erreur humaine à la défaillance organisationnelle : essai de mise en perspective historique – F.Guarnieri, J.Cambon, I.Boissières - - Mines Paris Tech, centre de recherche sur les risques et les crises – Revue REE (Revue de l'électricité et de l'électronique) n°8 – 09/2008*

[Hakima, 1998] *La Modélisation des Systèmes d'Information avec la méthode formelle B* – M.Hakima - Laboratoire Bases de données et systèmes d'information – Division systèmes d'information Cerist - RIST Vol.8 N°02 Année 1998

[Henry, 1994] *Contribution à l'évaluation de sûreté des logiciels de contrôle commande des centrales nucléaires - Application au SPIN N4* - J.Y. Henry, B. Soubiès, M. Le Meur, O.Elsensohn, J. Boulc'h - Congrès national de la SFEN - 1994

[Hadj, 2003] *La règlementation européenne en matière de sécurité ferroviaire* – H.Hadj-Mabrouk, I.Triki – Revue RGCF – 04/2003

[Hollnagel, 2004] *Resilience Engineering* – E.Hollnagel, D.Woods, N.Leveson – Aldershot : Ashgate – 2004

[Hollnagel, 2006] *Barriers and Accident Prevention* – E.Hollnagel – Aldershot : Ashgate – 2006

[Hollnagel, 2007] *Functional Resonance Accident Model (FRAM*" – E.Hollnagel– Pôle Cindynique, Sophia Antipolis, France – 2007

[Hollnagel, 2008] *Resilience Engineering Why, What and How*" – E.Hollnagel - École des Mines de Paris – France – 2008

[Hollnagel, 2009] *RAG – Resilience Analysis Grid – Technical Document prepared by the Industrial Safety Chair*" – E.Hollnagel - École des Mines de Paris – France – 01/2009

[Hoch, 2008] *Fiche événement de la Direction de l'Infrastructure – « Raté de fermeture PN23 et 22 de Hochfelden »* – Base REX SNCF - 02/2008

[Hungar, 2002] *Optimierung der Entwicklung bahntechnischer Systeme* - Werner Damm, Hardi Hungar, Bernhard Josko - Kuratorium OFFIS e.V. – 2002

[IDM, 2007] *„Mise en œuvre d'unités de preuve pour la vérification formelle de modèles“* – P.Dhaussy, J-C.Roger, F. Boniol / IDM 2007 – 04/2007

[IEC62278, 2002] IEC 62278 : *Railway Applications - The Specification and Demonstration of RAMS*", (2002).

[IEC62423, 2005] IEC 62425 Ed. 1: *Railway Application: Communications, Signalling and Processing Systems - Safety related Electronic System for Signalling*" -2005

[IEEE, 2003] *Software Development Worldwide : the State of the Practice*", Institute of Electrical and Electronics, Inc., IEEE Software, Vol 20 No 6, 2003

[IMdR, 2005] *Les normes 61508 et 61511 et leur implication* - Ministère de la recherche carré des Sciences Paris - IMdR – 05/2005

[IMdR, 2009] Projekt *„Apport des méthodes formelles pour les systèmes critiques* - APSYS E.Arberetrier, M.Antoni - IMdR - 01/2009

[IRJ, 2007] *British Track Faces Scruting after 152 km/h Derailment*, IRJ – 04/2007

[Iwata, 2008] *A study of evaluation methods for railway signalling systems from the viewpoint of availability* – K.Iwata – Railway technology avalanche N°25 p145 – 12/2008

[Jahnel, 2000] *Anwendung der EN bei kleineren Systemen und Anpassungsentwicklungen* – M.Jahnel, T.Koch, – SIGNAL +DRAHT 03/2000

[Jansen, 1999] *Referenzfallstudie Verkehrsleittechnik: Eine funkbasierte Bahnübergangsteuerung* – L.Jansen, E.Schnieder – 1999

[Kahn, 2008] *Sûreté de fonctionnement des systèmes programmes* – P.Kahn – Revue REE (Revue de l'électricité et de l'électronique) – REE n°8 – 09/2008

[Kassev, 2008] *An Application of phase–type Distributions for Modelling of Railway Safety-Critical-Systems* - K.Kassev, R.Slovák, E.Ivanov, N.Stoytcheva, E.Schnieder – FORMS08 – 10/2008

[Kinder, 2007] *Modelling and Formal Verification of Counting Heads for Railway* – S.Kinder, Rolf Dreschsler – Universität Bremen - FORMS07

[Klose, 2005] *Vermeidung systematischer Softwarefehler durch den Einsatz formaler Methoden für das ESTW B950* – J.Klose - ETR - Eisenbahntechnische Rundschau 03/2005

[Koren, 2005] *Translations between Textual Transition Systems and Petri Nets* – K.Korenblat, O.Grumberg, S.Katz - Computer Science Department The Technion - Haifa, Israel - {orna,katz}@cs.technion.ac.il

[Kotonya, 1998] *Requirements Engineering: Process and Techniques* – G.Kotonya, I.Sommerville – Wiley – 1998

[Lannoy, 1996] *Analyse quantitative et utilité du retour d'expérience pour la maintenance des matériels et la sécurité* – A.Lannoy - Édition Eyrolles, Collection de la Direction des Études et Recherches d'Électricité de France, 1996

[Laprie, 1984] *Dependability evaluation of software systems in operation* – J-C.Laprie – Laboratoire automatique analyses systèmes, Toulouse 31400 – IEEE transactions on software engineering 1984 Vol 10 n°6 - 1984

[Lartilleux, 1962] *Géographie des Chemins de fer français*- H. Lartilleux – Édition Chaix - 1962

[Lauber, 1998] *Prozessautomatisierung 1*- R. Lauber, P. Göhner, 3. Auflage, Springer Verlag, Berlin, Heidelberg, New York, 1998.

[Legof, 2009] Buch : *Sécurisation des architectures informatiques – Exemples concrets – Chapitre 3 De la TVM430 à l'architecture 2003* – Legof – Hermès Lavoisier – 2009 ISBN: 978-2-7462-1991-5

[Leveson, 2000] *The role of software in spacecraft accidents* - Pr N.G.Leveson – Aeronautics and Astronautics Dept. Massachusetts Institute of Technology – AIAA 2000

[Leveson, 2001] *Systemic factors in software-related spacecraft accidents* –Pr N.G.Leveson – Aeronautics and Astronautics Dept. Massachusetts Institute of Technology – AIAA 2001-4763 - 2001

[Lévi, 1988] *Problèmes de réutilisation liés au typage – Application à une extension du langage ADA“* – D. Lévi - Doktorarbeit – Université de Nice IMSP Lisan Spécialité Informatique - 1988

[Lévi, 1995] *Aérospatiale- Étude sur le soutien logistique du logiciel*" - DL/DL/P3.95.L.415 – 05/1995

[Litz, 2000] *Steuerungsentwurf mit Petrinetzen: Wichtige Nebensache* – L.Litz, G.Frey - IEE 45 – Jahrgang 2000 Nr. 2 – S.61-62 - 2000

[Löper, 2004] *Anforderungsdefinition und Anforderungsanalyse für sicherheitskritische Systeme (inkl. formale Methode)* – C.Löper - Vortrag im Rahmen des Seminars Analyse, Entwurf und Implementierung zuverlässiger Software -– Padelbron 01/2004

[Lötschberg, 2007] *Umfall im Lötschberg-Basislinie Tunnel mit ERTMS* - Eisenbahn Revue – 04/2008

[Lötschberg, 2008] E*TCS Software Error led to Derailment* - Railway Gazette International – 01/2008.

[Magnin, 2005-1] *Improved Algorithm for Computing Exact State Space of Petri Nets with Stopwatches* – M.Magnin, D.Lime, O.Roux - IRCCyN, Nantes, France - 2005

[Magnin, 2005-2] *An Efficient Method for Computing Exact State Space of Petri Nets with Stopwatches* – M.Magnin, D.Lime, O.Roux - Institut de Recherche en Communication et Cybernétique de Nantes - École Centrale de Nantes - Workshop on Software Model Checking - 2005

[Magnin, 2007] *Réseaux de Petri à chronomètres – Temps dense et temps discret*" – M.Magnin - École Centrale de Nantes - Doktorarbeit 12/2007

[Mahé, 2007] RGCF 165 – « *High Speed 1 : Le contrôle commande* » – Jean-François Mahé, Jöel Farhouat – 10/2007

[Marangé, 2008] *Vérification des propriétés de sécurité pour la commande des systèmes à événements discrets* – P.Marangé, F.Gellot, B.Riera - Centre de Recherche en STIC (CReSTIC), UFR des Sciences Exactes et Naturelles - 2008

[Marianom, 2006] *From UML to B - A Level Crossing Case Study"* - G.Marianom, J-L.Boulanger, P. Bon, , Computers in Railway pp. 351-362 – 2006

[Martin, 1990] *Le processeur code : un nouveau concept appliqué à la sécurité des systèmes de transport* – J.Martin, C.Galivel - Revue RGCF – 06/1990

[Mera ,2002] *Simulation of the ERTMS / ETCS Railways Control and Protection System; Levels 0, 1 and 2*" – J. M. Mera, L. M. Gutiérrez, et al. - 8th International Conference on Computer Aided Design, Manufacture and Operation in Railway and other Advanced Mass Transit Systems. COMPRAIL VIII. Lemnos, Greece – 06/2002.

[Méry, 2006] *Modélisation et développement de systèmes informatiques*" - Dominique Cansell, Dominique Méry, J-R Abrial - 09/2006

[Moens, 2007] *High Speed 1 : Le système de signalisation* – Gilbert Moens – RGCF 165 – 10/2007

[Monin, 1996] Buch : *Comprendre les méthodes formelles, Panorama et outils logiques* – J-F.Monin – Masson et CNET-ENST - 1996

[Monti, 2003] *SwAM ou méthode d'évaluation des logiciels appliqué à la grande vitesse (SwAM oder eine Methode zur Bewertung der bei der Hochgeschwindigkeit angewandten Software)* – A.Monti – Revue RGCF – 06/2004

[Mozoczi, 2007] *Possibilities of Validation of the Functional Requirements Specifications for Railway Interlocking and Interoperability – Questions and Considerations*" - L. Mosoczi – Hungarian State railways - FORMS07 – 01/2007

[Narboni, 2001] *Un cas remarquable de systèmes linéaires: les systèmes monotones. Résolution et application à la vérification formelle de programmes* – G-A. Narboni. - Doktorarbeit, Laboratoire Spécification et Vérification, ENS Cachan, France, 12/2001

[Nguyen, 2008] *Performance et sûreté des installations - impact des équipements programmés*" - Thuy Nguyen , Gilles Deleuze - EDF R&D – 3SGS08 – 03/2008

[Pachl, 2008] *Die Bedeutung betrieblicher Regelwerke für die Leit- und Sicherungstechnik* - Jörn Pachl – Signal +Draht – Rail Signalling and Telecommunication - 12/2008

[Pages, 1980] Buch : *Fiabilité des systèmes* – A.Pages, M.Gondran - Eyrolles – Collection des études et recherches de électricité de France – *11/1980*

[Parissis, 1996] *Techniques de test pour des logiciels réactifs synchrones - Testing Techniques for Synchronous Reactive Software* - Ioannis Parissis, Farid Ouabdesselam - Laboratoire de Génie Informatique - Institut IMAG Grenoble, France - 1996

[Park-Lee, 2005] *Performance Evaluation and Verification of Communication Protocol for Railway Signalling Systems* - G.T.Park, H.Lee, J. G. Hwang - Computer Standards & Interfaces, Volume 27, pp. 207-219 - 2005

[Patin, 2008] *Utilisation de la méthode formelle B pour un système SIL3 : la commande des portes palière sur la ligne 13 du métro Parisien / The B Formal Method for a SIL3 Sensor System: the Control System of the Platform Doors, Line 13 in Paris Subway*" - F.Patin, G.Pouzancre, D.Sabatiers, P. Sauvage (RATP) - 2008

[Pichon, 1886] *Note sur une solution générale des enclenchements ternaires* – L.Pichon (ingénieur principal de l'exploitation à la compagnie des chemins de fer du nord)– Revue RGCF (revue générale des chemins de fer) – 06/1886

[Pignal, 2003] *La base d'essais France pour ERTMS (Die Versuchsbasis Frankreich für ERTMS)* – O.Pignal, H.Thouvenot – Revue RGCF (Revue générale des chemins de fer et des tramways) – 12/2003

[Plan, 2008] *Modellbasierte Methoden – des Schlüssel zu Modularisierung und Standardisierung*" – O.Plan, T.Hiebenthal – Signal +Draht – Rail Signalling and Telecommunication – 11/2008

[PMI, 2004] *Preuve formelle de spécification – PMI* - Exposé ALCATEL – 04/2004

[Poncet, 2008] Diplomarbeit - *L'accident serait du à une erreur humaine* - Université de Paris X - Nanterre – F.Poncet – 09/2008

[Poole, 2003] *An Overview of Low Adhesion Factors for ERTMS* - Poole, W – 07/2003

[Prover, 2008] *Software and Services for Rail Control & Signaling Systems - Prover iLock Introduction* – Prover iLock – 01/2008

[Rakkay, 2005] *TSCPN: Timed Secure Colored Petri Net - Modélisation et Vérification de la sécurité des informations par des réseaux de Petri colorés temporisés*" – H.Rakkay - Doktorarbeit - École Polytechnique de Montréal - 08/2005

[Reason, 1990] Buch : *Human Error* – J.Reason – New York, NY : Cambridge University Press, 1990

[Reason, 1993] Buch : *L'erreur humaine* – J.Reason – Paris, Editions PUF, Collection « le travail humain », 1993

[Reason, 1995] *A system Approach to Organizational Error. Ergonomics* – J.Reason – vol 38 n°8 pp. 1708-1721, 1995

[Rétiveau, 1987] *La signalisation ferroviaire*", R.Rétiveau, Presses de l'école nationale des Ponts et Chaussées – 1987

[RFS, 1999] *Règles fondamentales de sûreté : Classement des matériels mécaniques, systèmes électriques, structures et ouvrages de génie civil*, IV.1.a, 1984, Logiciels des systèmes électriques classés de sûreté, II.4.1.a, 1999, Sûreté nucléaire en France, Les éditions des Journaux officiels

[RGCF168, 2008] *Procab : la solution d'un problème posé dès les années 1930 ?* - Clive Lamming – RGCF– 01/2008 (168)

[Rigaud, 2007] *La dialogique vulnérabilité résilience : une révolution scientifique pour les sciences et génies des activités à risques* – E.Rigaud - Pôle Cindyniques École des Mines de Paris – 2007

[RIL815, 2002] Hinweise des Fachautors zu Änderungen und Ergänzungen der RIL 815 – *Bahnübergangsanlagen planen und instandhalten* - 04/2002

[Roux, 2002] A *t-time Petri Net Extension for Real Time-Task Scheduling Modelling* - O.H. Roux, A-M. Déplanche – European Journal of Automation (JESA), 36(7), 2002

[Roux, 2004] *Time Petri Nets with Inhibitor Hyperarcs. Formal Semantics and State Space Computation* - O.H. Roux, D. Lime. - 25th International Conference on Application and Theory of Petri Nets, (ICATPN 2004), volume 3099 of Lecture - Notes in Computer Science, 371–390, Bologna, Italy, 06/2004. Springer-Verlag.

[Schlingloff, 2002] *Formale Methoden in der Praxis - Softwaredesign für Luft- und Raumfahrt – Antrittsvorlesung* – B.H. Schlingloff – Humboldt-Universität Berlin, Mathematisch-naturwissenschaftliche Fakultät II Institut für Informatik – 05/2002

[Schnieder, 1986] *Pro1992zessinformatik: Einführung mit Petrinetzen. Für Elektrotechniker und Informatiker, Maschinenbauer und Physiker nach dem Grundstudium* – E.Schnieder, Hrsg. - Friedrich Vieweg & Sohn, Programmserie „Regelungstechnik“, Braunschweig/Wiesbaden, ISBN 3-528-03358-4 - 1986

[Schnieder, 1992] Buch: *Petrinetze in der Automatisierungstechnik* – E.Schnieder, Hrsg. - R.Oldenbourg Verlag, München- 1992

[Schnieder, 2003] *International Workshop on Software Specification of Safety Relevant Transportation Control Tasks* - 23.-24. April 2002, Braunschweig. – E. Schnieder, Hrsg. - VDI Verlag, Fortschritt-Berichte VDI Reihe 12 Nr. 535, Düsseldorf - 2003

[Schnieder, 2007-1] *Verkehrsleittechnik: Automatisierung des Straßen- und Schienenverkehrs* – E.Schnieder - Springer Verlag, Berlin, ISBN 978-3-540-48296-3 - 2007

[Schnieder, 2007-2] *Methoden der Automatisierung: Beschreibungsmittel, Modellkonzepte und Werkzeuge für Automatisierungssysteme* - E. Schnieder, Hrsg - Programmserie „Automatisierungstechnik“ Vieweg & Sohn, Braunschweig/Wiesbaden - 1999

[Schnieder, 2008] *Toward Terminological Rigour in the Specification of Complex Automaton Systems* – L.Schnieder – Forms Format 2008 – 10/2008

[Schnieder, 2009] *Application of Formal Methods in Railway Signalling* – E.Schnieder, G.Tarnai Formal Methods for Automation and Safety - 2009

[Senesi, 2008] *Getting ERTMS into service quicker* – F.Senesi, G.Bonafe, S.Geradi, M.Frandli, N.Filipini – ATC project team, Rete Ferroviaria Italiana – Railway Gazette International – 12/2008

[Siemens, 2002] *B système - Spécifications fonctionnelles systèmes* - Journées du Groupe B du GDR ALP – Essame, D. Dolle, K. Doulaki, J. Falampin - CNAM : 06/2002

[Sirianni, 2004] *Modélisation, simulation et vérification de circuits numériques asynchrones dans le standard System C v2.0.1”* – Antoine Sirianni – Doktorarbeit INPG – 06/2004

[Slovak, 2003] *Profund modelling for holistic risk and avaibility analysis by means of stochastic Petri Nets applied to a level crossing control system* – R.Slovák, J.May, E.Schnieder - FORMS03 - 2003

[Slovak, 2005] *Potentials of Formal Modelling for Risk and Safety Analysis* - EURNEX Workshop on Level Crossing Safety, Braunschweig – Roman Slovak – 07/2005

[Slovak, 2007] *Formal structuring for development of level crossing ontology“* – R. Slovák, J. Drewes, L. Tordai, E.Schnieder – FORMS07 – 2007

[SNCF, 1941] *Règlement général de sécurité – Titre 1 Signaux / Traduction allemande* - SNCF 1941

[SNCF, 1945] *Annexe Est (ex-AL) au Règlement général de sécurité – Titre 1 Signaux / Traduction allemande* - SNCF 1945

[SNCF, 1948] *Auszüge aus des Dienstvorschrift P9a n°1– Sicherheit des Personals“* - Allgemeine Unfallverhütungsvorschriften – SNCF 1948

[SNCF, 1950] *Extraits du règlement P9a n°1 – Sécurité du personnel“* - Prescriptions générales à observer pour éviter les accidents – SNCF 1950

[SNCF, 1963] Buch : *La télécommande des postes d’aiguillages et Commande centralisée de la circulation* – A.Dheu - Direction des installations fixes – Édition 1961

[SNCF, 1963] Buch : *Cours d’enclenchements* – Direction des installations fixes – Édition 1963 – J.Walter – SNCF 1963

[SNCF, 1988] *Les circuits du PRS - Document d’étude à l’usage des agents chargés des essais“* - - VZE6 A. York - 1988

[SNCF, 1994] *Les bases de la sécurité – Édition 6* - IN0116 (EF 0 A3) - Document métier – 05/1994

[SNCF, 1995] *Étude de sûreté de fonctionnement permettant la comparaison entre les situations dégradées de SYMEL et de la voie unique CAPI/DAAT* – Étude SNCF-DTT (Direction des Transports Terrestres) – 04/1995

[SNCF, 1997] *Politique de maintenance des installations fixes et principales évolutions envisagées*, Direction de l’Équipement et de l’Aménagement, 12/1997

[SNCF, 1997] *Conditions techniques n°380 E pour la réalisation d’un compteur d’essieux NS1*“. CT EF 5B 31, 03/97 Direction de l’équipement et de l’aménagement

[SNCF, 2002] *Le poste informatique de technologie PC (PIPC) – Dispositions de principe – Généralités* - Référentiel Direction de l'Ingénierie - Directives d'Études de Signalisation – Édition du 29/08/2002

[SNCF, 2005] *Analyse du dérangement contraire à la sécurité au PAI de MELUN* – IGSF61 – 05/2005

[SNCF, 2008] *Dispositif de surveillance des points de comptage pour compteurs d'essieux NS1* – NT n°97/08 EF 5B 31

[SPR, 2008] *Software Quality in 2008 : a Survey of the State of the Art*", SRP Software Productivity Research LLC – JaSST Japan Symposium on Software Testing, 2008

[ST07, 2007] Proceedings of the 5th Workshop on *„Systems Testing and Validation (ST07)*" – 12-2007 – ISBN 978-3-8167-7475-4

[Staffelbach, 2008] *Each time you change a bit or byte in your system, you have to run through the whole process again – Dr T.Staffelbach – head of Train Protection, SBB Infrastructure – Terrapinn's EuroRail 2008 – Milano – 02/2008*

[SYSTRA 2004] *Technique ferroviaire – Formation approfondie* – SYSTRA - Réseau Ferré de France 2004 – Réf : 4071/FRA3/PAR/906-03 – Édition 2 – du 16/01/2004

[Tarnai, 2001] *Zusätzliche Aspekte zur Anwendung von formalen. Techniken in der Eisenbahnsicherungstechnik* – G.Tarnai, B.Sághi - Signal+Draht - 8/2001.

[Tarnai, 2006] *Proceedings of Formal Methods for Automation and Safety in der Eisenbahnsicherungstechnik* - Signal + Draht. 98(6), S. 6-10, 2006.- E.Schnieder, G.Tarnai - 2006

[Tarnai, 2009] *Az MSc képzés programja – az egészégügyi mérnök, a mérnök informatikus és a villamosmérnöki szakokon* – V2.5.4 – Budapesti Müszaki ès Gazdasàgtudomànyi Egyetem Villamosmérmöki ès Informatikai Kar - 2009

[Terada, 2002] *Application of Formal Methods to the Railway Signalling Systems*, Natzuki Terada, Mitsuyoashi Fukuda – QR of RTRI, Vol 43, No4, 12/2002

[Terada, 2008] *Application of Formal Methods to the Railway Signalling Systems* – FORMS 2008 - Natsuki Terada, Mitsuyoshi Fukuda - 2008

[TSI94, 1994] *Automates concurrents à contraintes* - L.Fribourg, M.Veloso Peixoto - Technique et Science Informatiques 13(6), pages 837-866, 1994. http://www.lsv.ens-cachan.fr/Publis/PAPERS/PS/LF-TSI94.ps

[UIC, 2007] *Generic Hazard List Methodology for Railway Signalling* – UIC Copyright – ISBN 2-7461-1264-7 (English issue) – 02/2007

[UKHSE, 1995] *Out of control – Why Control Systems Go Wrong and How to Prevent Failure* - UK HSE Books, 1995

[Vallée, 1998] Buch : *Study of the Test Contribution to Software Reliability Quantitative Assessment for Nuclear Safety-Critical Systems* - F.Vallée, J.-Y.Henry, F.Daumas - ISSRE Proceedings - 1998

[Villemeur, 1980] Buch : *Sûreté de fonctionnement des systèmes industriels – Fiabilité – Facteurs humains – Informatisation* – A.Villemeur - Eyrolles – Collection des études et recherches de électricité de France – *03/1997*

[Villemeur, 1992] *Les méthodes de la fiabilité et la sûreté de fonctionnement*, T. Desmas, C.Ancelin, A.Villemeur - Revue Générale de l'Électricité, n° 8/92 - 09/1992

[Vogt, 2006] *Einführung in die Informatik II - Objektorientierte Modellbildung* - 5.5.2006 - F.Vogt, Dr. M.Venzke - Technische Universität Hamburg-Harburg Arbeitsbereich Telematik

[Watteyne, 2005] *Proposition et validation formelle d'un protocole MAC temps réel pour réseaux de capteurs linéaires sans fils* - Thomas Watteyne Diplomarbeit Informatique Spécialité Réseaux, Télécommunications et Services 2004/2005 – Laboratoire CITI – INSA de Lyon

[Zoller, 2002] *Handbuch des ESTW Funktionen – die Sicherungsebene im elektronischen Stellwerk* - H. J. Zoller: - 1.Auflage, Tetzlaff Verlag, Hamburg, 2002

[L108/4, 2009] *Commission Regulation (EC) No 352/2009 of 24 April 2009 on the Adoption of a Common Safety Method on Risk Evaluation and Assessment as Referred to in Article 6(3)(a) of Directive 2004/49/EC of the European Parliament and of the Council (Text with EEA Relevance)* – 20.4.2009

[2004R0881] *Regulation (EC) No 881/2004 of the European Parliament and of the Council of 24 April 2004 Establishing a European Railway Agency* – 2004R0881-EN-01.01.2009-001.001-1

[2004L0049] *Directive 2004/49/EC of the European Parliament and of the Council of 29 April 2004 on Safety on the Community's Railways and Amending Council Directive 95/18/EC on the Licensing of Railway Undertaking and Directive 2001/14/EC on the Allocation of Railway Infrastructure Capacity and the Levying of Charges for the use of Railway Infrastructure and Safety Certification* - Railway Safety Directive – 2004L0049-EN-24/12/2008-001.001-1

**Résumé**

Le développement de l'informatique et celui de l'automatisation ont été et sont encore sources de nouvelles solutions de plus en plus efficaces, mais sont aussi sources de nouvelles complexités rendant encore plus difficiles la conception durable et économique, l'évaluation, de la sûreté de fonctionnement des équipements et des systèmes de transport. L'informatique temps réel est maintenant présente dans les systèmes gérant les vies humaines. Il apparaît maintenant que les méthodes et normes actuelles ne permettent pas toujours de répondre aux attentes en matière de disponibilité et de sécurité. Ainsi les erreurs systématiques des logiciels sont à craindre. Le travail a consisté à définir et instrumenter une méthode de conception et de validation. Le travail montre qu'il est possible d'appliquer sur des automatismes industriels une méthode de validation formelle. Le cas des automatismes ferroviaires est plus particulièrement traité. La méthode proposée repose sur plusieurs actions de conception:

- tenir le plus grand compte du contexte métier, d'identifier les propriétés de sécurité et les postulats de fonctionnement ;
- distinguer les logiciels fonctionnels et les logiciels de base (gestion du matériel et interprétation des fonctions métier) ;
- spécifier les fonctions sous forme d'automates écrits en langage AEFD. Ce langage permet une écriture des réseaux de Petri et une interprétation déterministe.

Dans ces conditions il s'avère possible de réaliser une validation formelle d'un automatisme ferroviaire, un poste d'aiguillage par exemple.

L'idée principale consiste à développer un automate industriel de sécurité qui se comporte comme une machine abstraite (un automate concurrentiel à contraintes et à temps de transition nul) afin d'en permettre une validation formelle ultérieure. L'écriture des graphes fonctionnels s'adresse à des personnes possédant une compétence métier sans aucune connaissance particulière en informatique.

Les réseaux de Petri sont un langage de conceptualisation. Nous avons utilisé ces réseaux, écrits en langage AEFD, comme langage de spécifications interprétables. Les propriétés de sécurité et les postulats de fonctionnement sont été écrits de la même manière. La méthode proposée permet dans ces conditions de réaliser une preuve formelle des fonctions du système.

**Kurzfassung**

Die Entwicklung der Informatik und der Automatisierung ist (und war schon immer) Quelle stets neuer und effizienterer Lösungen, aber auch neuer Komplexität; sie macht eine dauerhafte, wirtschaftliche Gestaltung und Überprüfung der Sicherheit der Anlagen und der Verkehrssysteme nur noch schwieriger. Die Echtzeitinformatik ist heutzutage in die Systeme integriert, die das Leben von Menschen verwalten. Es stellt sich heraus, dass die derzeitigen Methoden und Normen nicht immer den Anforderungen nach Verfügbarkeit und nach Sicherheit entsprechen. Es sind so systematische Fehler der Software zu befürchten. Die vorliegende Arbeit besteht darin, eine Konzeptions- und Überprüfungsmethode zu definieren und zu instrumentalisieren. Sie zeigt, dass es möglich ist, formale Überprüfungsmethoden auf industrielle Steuerungen anzuwenden. Die Eisenbahnsteuerungen werden besonders behandelt. Die vorgeschlagene Methode beruht auf mehreren Konzeptionsideen:

- so weit wie möglich das berufliche Umfeld berücksichtigen, die Sicherheitseigenschaften und die Funktionsanforderungen identifizieren
- die Funktionssoftware und die Grundsoftware (Verwaltung des physikalischen Materials und Interpretation der fachspezifischen Aufgaben) unterscheiden
- die Funktionen in Form von Automaten in AEFD-Sprache schreiben. Diese Sprache erlaubt eine Formulierung von Petrinetzen und eine deterministische Interpretation.

Unter diesen Bedingungen ist es möglich, eine formelle Überprüfung einer Eisenbahnsteuerung, z.B. eines Stellwerks, zu verwirklichen.

Die Hauptidee besteht in der Entwicklung eines industriellen Sicherheitsautomaten, der sich wie eine abstrakte Maschine verhält, damit dieser später formal validiert werden kann. Das Hilfsprogramm für die Formulierung der funktionellen Graphen ist für Fachleute bestimmt, die nicht über besondere Informatikkenntnisse verfügen.

Die Petrinetze sind eine Konzeptualisierungssprache. Diese Netze, die in der AEFD-Sprache formuliert sind, werden als interpretierbare Spezifikationen benützt. Die Sicherheitseigenschaften und die Anforderungen sind auch auf dieselbe Art formuliert. Die vorgeschlagene Methode erlaubt es unter diesen Bedingungen, einen formalen Beweis der Funktionen des Systems durchzuführen.